Visual Support for Safety Analysis

Vom Fachbereich Informatik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation
von

Yi Yang

Datum der wissenschaftlichen Aussprache: 25. Oktober 2012

Dekan: Prof. Dr. Arnd Poetzsch-Heffter
Vorsitzender der Prüfungskommission: Prof. Dr. Stefan Deßloch
Erster Berichterstatter: Prof. Dr. Peter Liggesmeyer
Zweiter Berichterstatter: Prof. Dr. Hans Hagen

D 386

Abstract

The safety of embedded systems is becoming more and more important nowadays. Fault Tree Analysis (FTA) is a widely used technique for analyzing the safety of embedded systems. A standardized tree-like structure called a Fault Tree (FT) models the failures of the systems. The Component Fault Tree (CFT) provides an advanced modeling concept for adapting the ordinary FTs to the hierarchical architecture model in system design. Minimal Cut Set (MCS) analysis is a method that works for qualitative analysis based on the FTs. Each MCS represents a minimal combination of component failures of a system called basic events, which may together cause the top-level system failure. The ordinary representations of MCSs consists of plain text and data tables with little additional supporting visual and interactive information. Importance analysis based on FTs or CFTs estimates the contribution of each basic event to a top-level system failure. The resulting importance values of basic events are typically represented in summary views, e.g., data tables and histograms. There is little visual integration between these forms and the FT (or CFT) structure. The safety of a system can be improved using an iterative process, called the safety improvement process, based on FTs taking relevant constraints into account, e.g., cost. Typically, relevant data regarding the safety improvement process are presented across multiple views with few interactive associations. In short, The ordinary representations cannot effectively facilitate these analyses.

We propose a set of visualization approaches for addressing the issues above mentioned in order to facilitate those analyses in terms of the representations.

Contributions:

- To support the MCS analysis, we propose a matrix-based visualization that allows detailed data of the MCSs of interest to be viewed while maintaining a satisfactory overview of a large number of MCSs for effective navigation and pattern analysis. Engineers can also intuitively analyze the influence of MCSs of a CFT.
- To facilitate the importance analysis based on the CFT, we propose a hybrid visualization approach that combines the icicle-layout-style architectural views with the CFT structure. This approach facilitates to identify the vulnerable components considering the hierarchies of system architecture and investigate the logical failure propagation of the important basic events.
- We propose a visual safety improvement process that integrates an enhanced decision tree with a scatter plot. This approach allows one to visually investigate the detailed data related to individual steps of the process while maintaining the overview of the process. The approach facilitates to construct and analyze improvement solutions of the system safety.

Using our visualization approaches, the MCS analysis, the importance analysis, and the safety improvement process based on the CFT can be facilitated.

Zusammenfassung

Sicherheit ist ein zentraler Begriff in der heutigen Entwicklung von komplexen eingebetteten Systemen. Die Fehlerbaumanalyse ist eine weit verbreitete Technik, mit deren Hilfe gefährliche Systemausfälle, unter Verwendung einer standardisierten baumartigen Struktur, genannt Fehlerbaum, modelliert und analysiert werden können. Der Komponentenfehlerbaum stellt eine Weiterentwicklung dieses Konzeptes dar. Er ergänzt die traditionellen Ansätze um zusätzliche Aspekte, welche es erlauben Fehlerbäume entsprechender Analyse zugrundeliegenden Systemarchitektur zu modularisieren. Als Komplement der Fehlerbaumanalsye erlaubt die Minimal Cut Set (MCS) Analyse die detailierte Betrachtung möglicher Ausfallszenarien mit Hilfe sogenannter Minimaler Schnittmengen (im englischen Minimal Cut Sets genannt). Die standardisierten Darstellungsvarienten von MCS sind vorwiegend listen- und tabellenbasiert, die in ihrer ursprünglichen Form nur wenig Anhaltspunkte für detailliertere Analysen bieten. Basierend auf der (Komponenten-) Fehlerbaumanalyse schätzt die "Importance Analysis" hingegen die Beiträge von einzelnen Teilsystemausfällen zum sicherheitskritischen Gesamtsystemausfall ab. Die Ergebnisse werden in der Regel mittels aggregierten Darstellungen, bestehend aus mehreren Tabellen- und Histogrammansichten untersucht und bewertet. Es gibt nur wenig visuelle Korrelationen zwischen diesen unterschiedlichen Repräsentationen. Die Sicherheit eines Systems kann in einem iterativen Prozess verbessert werden. Der Sicherheitsverbesserungsprozess basiert auf Berücksichtigung der einschlägigen Beschränkungen, wie z.B. den Kosten. Die relevanten Daten bezüglich des Sicherheitsverbesserungsprozesses sind meist über unterschiedliche Ansichten verteilt und entkoppelt und für detaillierte Untersuchungen wenig geeignet.

Beiträge:

- Es wurde eine matrixbasierte Visualisierung für die MCS Analyse entwickelt, die es gleichzeitig erlaubt, die Musteranalyse und die Untersuchung der detaillierten Daten bezüglich der Zusammensetzung der MCSs vorzunehmen. Die Einflüsse von MCSs können intuitiv analysiert werden.
- Im Rahmen der Importance Analysis wird eine Visualisierung präsentiert, die die Architektursichten im Icicle Layout mit der Struktur des Komponentenfehlerbaums kombiniert. Die Visualisierungsmethode erlaubt die Untersuchung der wichtigen Systemkomponenten und die Ausbreitung der einzelnen kritischen Teilsystemausfälle zu analysieren.
- Die entwickelte Methode zur Visualisierung eines Sicherheitsverbesserungsprozesses integriert einen erweiterten Entscheidungsbaum mit einem Scatter Plot. Die Methode erlaubt die detaillierten Informationen über einzelne Schritte des Verfahrens hinweg zu untersuchen und trotzdem den Überblick über den Prozess zu behalten. Die Methode erleichtert Lösungen zur Verbesserung der Sicherheit eines Systems zu konstruieren und zu analysieren.

Acknowledgments

Many people gave me great help when I was working on my PhD. I would like to take this chance to thank them. First of all, I am grateful to my advisors Prof. Peter Liggesmeyer and Prof. Hans Hagen. They gave me an opportunity to work in a research domain that I am interested in. I wish to thank them for giving me a chance to be a member of the International Research Training Group "Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling, and Engineering" (IRTG 1131). The advisors provided me with a good research environment and their guidance which ensured the correct research direction.

I would like to thank Dr. Patric Keller and Dr. Dirk Zeckzer whom I worked with in the period of my doctoral study. Their constructive suggestions greatly helped me to overcome the confusions in my study. I wish to thank Prof. Charles Hansen who gave me considerable support when I was visiting the University of Utah. I would like to thank Dr. Yarden Livnat who gave me many highly valuable advices which greatly facilitated my research work in Utah.

I would like to thank the colleagues of the Software Engineering Research Group: Dependability, the Computer Graphics and HCI Group, and the Software Engineering Research Group: Processes and Measurement, at the University of Kaiserslautern, as well as colleagues working at the Scientific Computing and Imaging Institute (SCI Institute) at the University of Utah, and the colleagues at Fraunhofer IESE. In particular, I would like to thank Yasmin I. Al-Zokari, Taimur Khan, and Sven Böttger. Our early cooperation made the foundation of this dissertation. I would like to give special thanks to Mady Gruys, Inga Scheler, Kristina Jerkku, Thomas Schneider, Roger Daneker, and Deborah Zemek. They helped me to manage organizational issues in Kaiserslautern as well as in Utah. Furthermore I would like to thank Peter Walton, Kavyashree Jamboti and Max Steiner for the proof-reading and all people that supported and promoted my work.

Finally, I would like to give my deepest gratitude to my grandfather and my parents. Their love gave me the confidence and courage to get through difficult times. Without the constant support of my family, this dissertation would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Today, embedded systems are very popular in everyday life. Safety plays an important role in the design of embedded systems. Safety refers to a state of a system where the danger to a person or property is below an acceptable value [10, 92, 96]. An embedded system is treated as a safety-critical system whose failure might lead to unacceptable consequences and endanger human life, substantial economic loss, or cause extensive environmental damage [118]. Common examples of safety-critical systems are cars, medical equipments, airplanes, and nuclear power plants. Safety analysis is often referred to as being a process whose goal is to provide a reliable assessment of the risks of a system. Fault Tree Analysis (FTA) is a widely used safety analysis technique that deductively constructs tree-like models called Fault Trees (FTs). In order to effectively adapt FTs to the system architectural model, an advanced modeling concept called Component Fault Trees (CFTs) was proposed. Sub-trees of a FT are modularized as a CFT component according to system architectural components.

There are two main categories of analyses based on FTs (also CFTs): qualitative analysis and quantitative analysis. Minimal Cut Set analysis (MCS analysis) is an important method of qualitative analysis. Each MCS consists of a minimal combination of the failures of components (called Basic Events) that may together cause a specific system failure. The MCS analysis provides all possible scenarios (MCSs) that individually lead to the specific undesired failure. The importance analysis and sensitivity analysis are commonly used quantitative analyses of the FTA. The importance analysis investigates the respective contributions of basic events to the undesired system failure. Additionally, in order to identify the most appropriate improvement solution(s) for the system design with respect to safety, engineers usually apply an iterative safety improvement process.

There are few suitable visualization methods to represent the results of these analyses. Even though a great deal of effort has been vested in novel methods for the analyses, there has been very little contribution to the representations of the results. The representation concepts of the MCS analysis consist of plain text and tables with few supporting visualizations and interactions. The representation concepts of the importance analysis concentrate on the ordinary data-aggregated forms, such as

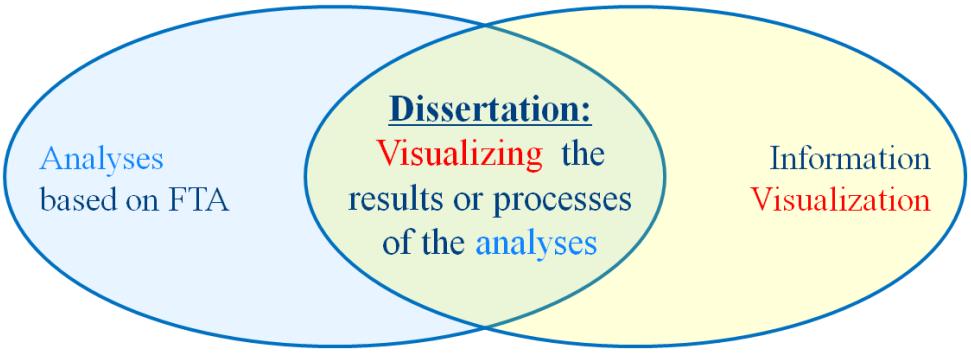


Figure 1.1: The research area of the dissertation. We propose concepts that facilitate the FTA with the help of information visualization techniques.

tables, histograms, and pie charts. For the safety improvement process, the related data are usually represented over separate views (most of them are tables) without visually and interactively associating with the decision process of the improvements. In this way, engineers need to manually look up the desired data in separate views.

To sum up, the currently applied representation concepts of the MCS analysis, the importance analysis, and the safety improvement process require a great deal of additional effort and reduce the effectiveness of the analyses because of the lack of the appropriate representations. We aim to facilitate these safety analyses by using the suitable visualization methods and interaction techniques. Figure 1.1 illustrates the research area of the dissertation. In this chapter, the contributions of the work and the structure of the dissertation are briefly described.

1.2 Contributions

For the MCS analysis, we propose an enhanced matrix-based visualization that has the following benefits:

- satisfactory overview of a large number of MCSs while investigating detail information;
- effective highlighting concepts for quickly exploring the significant information;
- integrated representation of failure propagation for analyzing the influence of MCSs along the CFT structure.

Our matrix view facilitates the identification of the important information from large-scale MCS data and the analysis of the effect of MCSs along CFT structure.

To enhance the representation of the quantitative importance analysis, we propose a hybrid visualization approach integrating the icicle layout with the CFT structure. This approach has the following advantages:

- clear and concise representation for exploring the result of the importance analysis by taking the hierarchical model of the system design into account;
- flexible representations and interactions for investigating the failure propagation of the important basic events;
- adaptable view for analyzing the importance of a large number of basic events.

Our visualization works on identifying the critical CFT components according to the important basic events as well as analyzing the failure propagation of the important basic events.

In order to support the safety improvement process, we propose an approach that visualizes and integrates the significant data relevant to the safety improvement process using a combination between an enhanced decision tree and a scatter plot. The benefits of the visualization approach focus on the following aspects:

- visual representations of the significant data for intuitively analyzing the data;
- a satisfactory overview of the sequential design modifications while investigating the detailed data of specific modifications;
- flexible visual context of the logical CFT structures adapting to the procedure of the safety improvement process.

Our approach facilitates the safety improvement process by visualizing and integrating graphical properties of significant data.

1.3 Content

Chapter 2 introduces the fundamentals of safety analysis and the essentials of information visualization, as well as the previous work including the currently used representation concepts for safety analysis and the visualization techniques related to our work. The motivations and the application examples depicting the motivations are described in Chapter 3. Chapter 4 conceptually introduces the design concepts of our visualization approaches. Chapter 5 presents the matrix-based visualization for the MCS analysis. Visualization methods for visually enhancing the importance analysis are introduced in Chapter 6. The visualization approaches for facilitating the safety improvement process are proposed in Chapter 7. Chapter 8 briefly describes the tool environment of *ViSSaAn* that was implemented according to the visualization approaches introduced in the previous chapters. Finally, a general conclusion is provided for the proposed visualization concepts and the future work is discussed in Chapter 9.

Chapter 2

Background and Related Work

2.1 Safety and Reliability Analysis

The fundamentals of safety and reliability analysis are introduced in this section. The section is based on the lecture of "Safety and Reliability of Embedded Systems" conducted by Prof.Liggesmeyer at the University of Kaiserslautern [126].

2.1.1 Basic Concepts

2.1.1.1 Quality

Quality is defined as "*a degree to which a set of inherent characteristics fulfills requirements*" by ISO9000 [98]. The quality of a system is assessed by evaluating the quality characteristics of the system [56, 96, 98]. A characteristic is a distinguishing feature [98], e.g., reliability of a car.

2.1.1.2 Failure and Fault

IEC 61508 [92] defines the terms *failure* and *fault* as follows:

- Failure: "*termination of the ability of a functional unit to perform a required function.*"
- Fault: "*abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.*"

2.1.1.3 Reliability

Reliability is defined as "*the probability that an item can perform a required function under given conditions for a given time interval*" by IEC 60050 [90]. Reliability is an important quality characteristic of embedded systems. The life distribution of a system $F(t)$ is the probability that lifetime T is less or equal to time t. It means that a system has already failed by t.

$$F(t) = P(T \leq t)$$

There are the following assumptions for the distribution:

- $F(t = 0) = 0$, i.e., the system is intact.
- $\lim_{t\to\infty} F(t) = 1$, i.e., system sometimes fails.

Reliability $R(t)$ is the probability that at time t no failure has occurred:

$$R(t) = 1 - F(t)$$

The life distribution $F(t)$ is also called the *unreliability*, *failure probability*, or *probability of failure*.

2.1.1.4 Risk

Risk is defined as "*combination of the probability of occurrence of harm and the severity of that harm*" by IEC 61508 [92]. The frequency of harm may be the failure probability or failure rate.

$$Risk = H * S$$

, where H is the expected frequency of an event that may cause harm, and S is the expected severity of the harm.

2.1.1.5 Safety

Safety is defined as a state of a system that is "*freedom from unacceptable risk*" by IEC 61508 [92].

2.1.1.6 Safety Analysis

Safety analysis is a process that evaluates the risks of a system and aims at the acceptance of those risks [119, 124].

2.1.1.7 Probabilistic Risk Assessment (PRA)

Probabilistic Risk Assessment (PRA) is a logical safety analysis method that evaluates risks of the complex systems using probability theory [140,142]. NASA [140,142] illustrates the typical task flow of PRA (Figure 2.1) and depicts the task flow as follows:

- **Objectives definition.** The goals of the PRA and the undesirable consequences need to be determined.
- **System familiarization.** The system must be known well with respect to the design and operational information.
- **Initiating events identification.** All initiating events that may cause the defined undesirable consequences must be identified.
- **Structuring scenarios.** An accident scenario describes a case that an initialing event causes an undesirable consequence. Using the inductive Event Tree (ET), each accident scenario is modeled as a series of events that starts with an initiating event and has sequentially caused intermediate events as well as a final undesirable consequence.

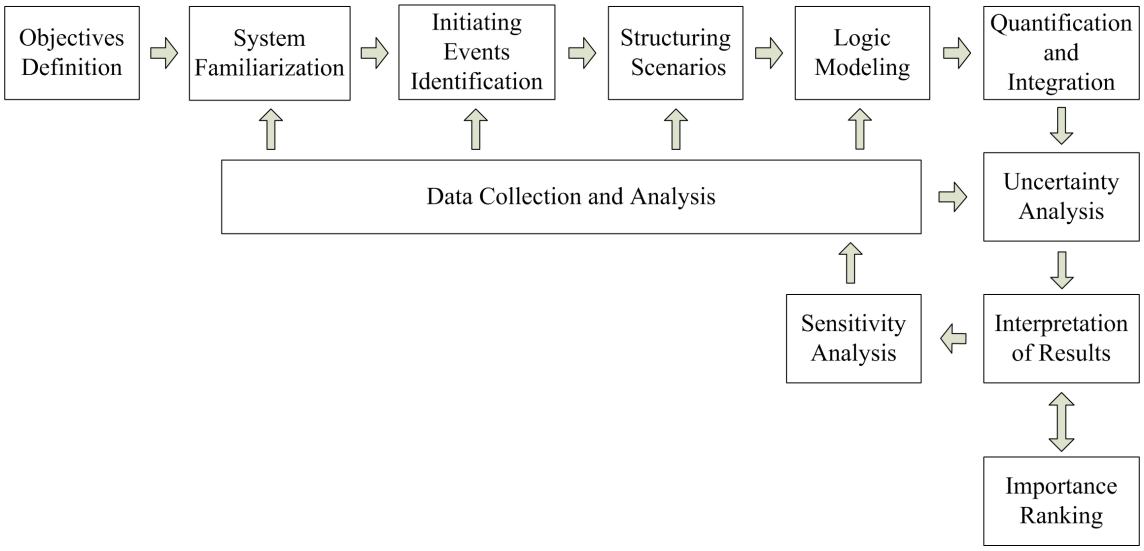


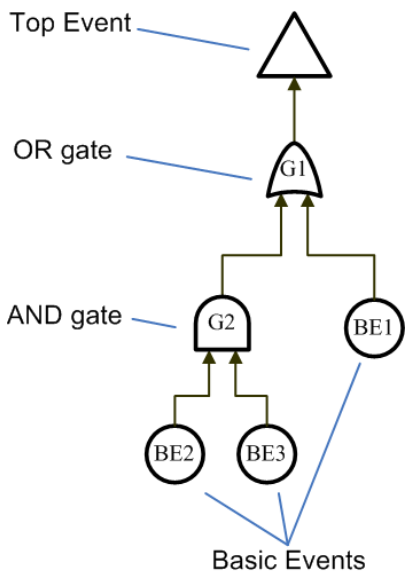
Figure 2.1: Typical task flow of PRA [142].

- **Logic modeling.** For a specific intermediate event in an accident scenario, the failures of this event are modeled as a Fault Tree (FT) (section 2.1.2). A FT illustrates how the given event is caused by a set of logically connected basic events. The constructed FTs are connected as a whole model for quantifying the accident scenarios.
- **Data collection and analysis.** The data that is used in the PRA process must be collected and processed.
- **Quantification and integration.** The constructed FTs are logically connected and quantified for determining the frequency of occurrence of the undesirable consequences.
- **Uncertainty analysis.** Uncertainty analysis measures the confidence in the quantification results based on FTs. The commonly used method is Monte Carlo simulation.
- **Sensitivity analysis.** Sensitivity analysis measures the relations between the changes of the inputs of PRA and the resulting impacts on the risk.
- **Importance ranking.** Importance ranking (importance analysis) identifies the major failures that contribute the most to risk.

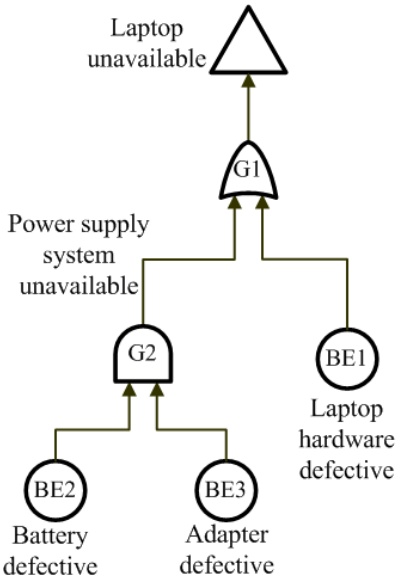
2.1.1.8 Risk Acceptance and Risk Reduction

Risk Acceptance represents that the actual risk is below or equal to the specific acceptable value. It provides a decision whether the risk of a system or a component is acceptable, i.e., whether the system or the component is safety. The commonly used methods are MEM (Minimal Endogenous Mortality) [37], GAMAB (Globalement Au Moins Aussi Bon) [37], ALARP (As Low as Reasonably Practicable) [88], and SIL (Safety Integrity Level) [92].

Risk Reduction identifies ways to reduce the potential loss associate with the actual risks. It is used to achieve the goal of the safety analysis. Risk acceptance and risk reduction are important strategies in the risk management [99].



(a) A FT consists of three basic events that are connected by an AND-gate and an OR-gate.



(b) FT with the semantic meaning.

Figure 2.2: Fault Tree.

2.1.2 Fault Tree Analysis

Fault Tree Analysis (FTA) is an effective technique of safety and reliability analysis for embedded systems. The FTA was proposed for the Minuteman Launch Control System in 1961 [63], and then defined in international standards [55, 91, 92, 142, 143]. This is a deductive analysis method allowing to trace the causes of an undesired system state back to its roots. The FTA is based upon the usage of so-called Fault Trees (FTs). The FTA contains qualitative analysis and quantitative analysis for FTs. The FTA is mainly applied to the following aspects [142]:

- understanding of the logic leading to a specific failure.
- analysis of the possible scenarios of the low-level failures that sufficiently cause a specific failure.
- priority of the failures leading to a specific failure.
- optimization of resources.
- improvements of the vulnerabilities of the system safety.

2.1.2.1 Fault Tree

Fault Tree (FT) is the core of the FTA (Figure 2.2 (a)). A FT is represented as a tree-like structure that consists of minimal three types of basic elements: Top Event, Basic Event, and Gate.

- The *top event (TE)* is the root (output) of the FT and represents the specific failure of a system.

- The *basic events (BEs)* are leaf nodes of the FT and represent the failures that may cause a top event to occur. A basic event will no longer be refined. Engineers may assign a failure probability to a basic event. The failure probability of a top event can be calculated according to the failure probabilities of its basic events. There are variants of basic events including Conditioning Events, Undeveloped Events, and House Events.
- The *gates* are the intermediate nodes of the FT that logically connect elements of the FT. There are various types of gates that represent different logical relations among elements. The basic types of gates are the OR-gate and the AND-gate. The OR-gate represents that its output event occurs only if at least one of the input events occurs. The AND-gate represents that its output event occurs only if all input events simultaneously occur. Other types of gates, e.g., the Priority-AND gate and the Exclusive-OR gate, are variants of the AND-gate and the OR-gate [142].

Additionally, there are two kinds of elements for modularization of the FT: *Transfer In* and *Transfer Out*. There are two styles for the graphical symbols of the FT: US [142] (Figure 9.1) and European [91]. Figure 2.2 (a) shows an example of the FT. There are three basic events "BE1", "BE2" and "BE3". The basic events "BE2" and "BE3" are connected by an AND-gate "G2". The basic event "BE1" is connected with the failure coming from "G2" by an OR-gate "G1". The failure coming out from "G1" is represented by the top event.

Besides qualitatively representing relations among elements of FTs, gates can also perform boolean calculation for the output failure probabilities. The failure probability of the AND-gate P is the product of failure probabilities of all input elements p_i. The probability is formulated as

$$P = \prod_{i=1}^{n} p_i$$

The failure probability of the OR-gate P is formulated as

$$P = 1 - \prod_{i=1}^{n} (1 - p_i)$$

The FTA provides a deductive method to construct a FT starting at a specific undesired failure, i.e., the top event. The FTA iteratively determines the causes of the specific failure, i.e., basic events. In the process, the FTA uses gates to connect the causes found. A FT represents the logical interrelationships of basic events that cause the specific failure, i.e., the top event of the FT. Figure 2.2 (b) shows a FT that has semantic meaning. The top event of the FT represents the failure "Laptop is not available". This problem may be caused either by the failure of "Laptop hardware is defective (BE1)" or by the failure "Power supply system is not available (G2)" because both failures are connected by the OR-gate "G1". The failure "Power

supply system is not available (G2)" may be caused when the failures of "Battery is defective (BE2)" and "Adapter is defective (BE3)" simultaneously occur because they are connected by the AND-gate "G2". In this way, the basic failures (i.e., basic events) that may cause the top event are identified and their logical relations are represented by logical gates.

After generating a FT, it still needs the suitable methods to quantitatively and qualitatively analyze the FT. Qualitative analysis focuses on the combinations of basic events that may lead to the top event. Quantitative analysis works on investigating the failure probability of the top event, and the estimation of basic events.

A FT may be large when the analyzed system is complex. Ericson et al. [94] provided a rule: small FT (<100 events), medium FT (100 to 1,000 events), and large FT (> 1,000 events). The events include basic events, variants of basic events (e.g., House Events), and gates (as intermediate events).

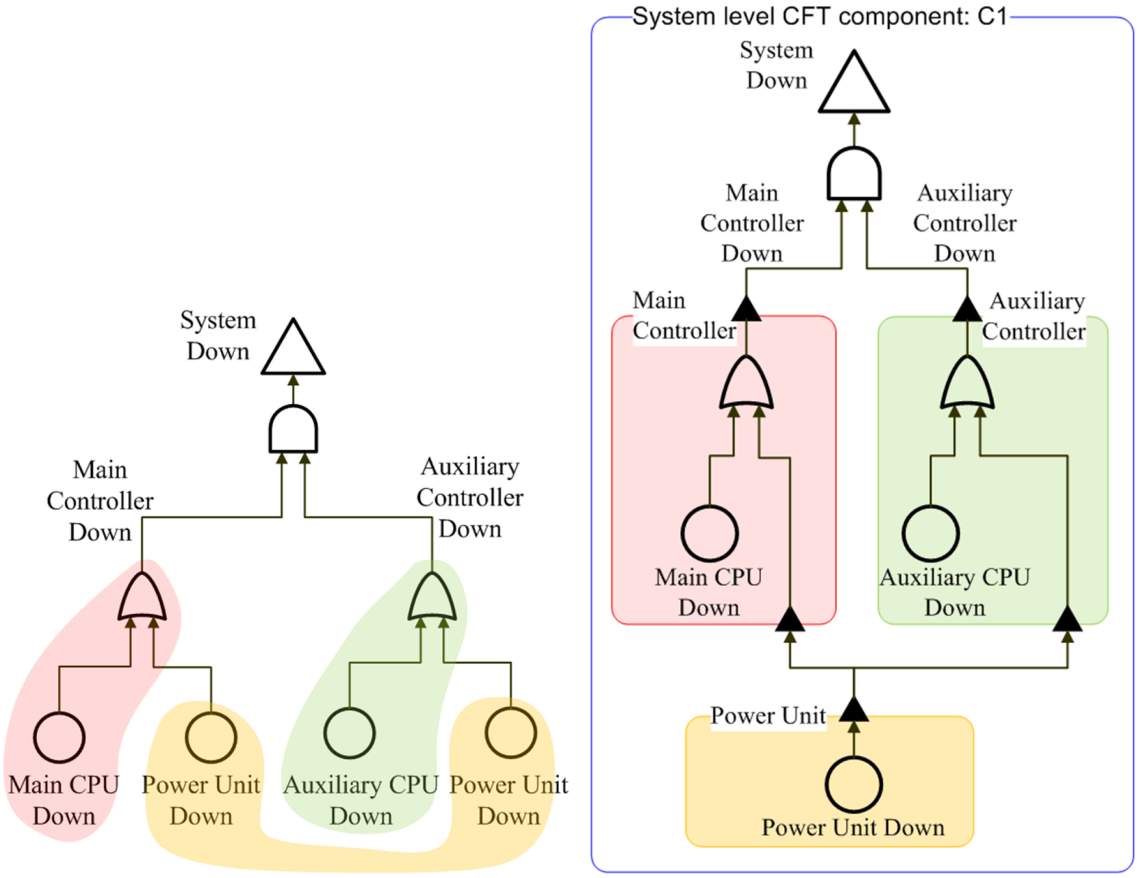
2.1.2.2 Component Fault Tree

For modeling a complex system in the design phase, engineers usually divide the system into architectural components (i.e., technical components) by applying a hierarchical decomposition concept to the system architecture [43, 76]. The system architectural components may be recursively divided into sub-components.

The traditional modularization concept allows FTs to be organized according to the contained sequence of failure influences into different *modules*, where each module represents an independent sub-tree [55, 108]. However, this concept does not allow FT modules to be used the same way as is possible for system architectural components in the system design. Additionally, because a failure may influence the top event in multiple scenarios, engineers usually build multiple basic events for the failure over different sub-trees of the FT. In this way, the "repeated events" issue is introduced in the FT construction.

To solve these issues, Kaiser et al. [107, 108] proposed an extended modeling concept called *Component Fault Trees (CFTs)* by taking the hierarchical system architecture into account. With respect to safety analysis, a system can be modeled using CFT components instead of modules. Figure 2.3 demonstrates an example with respect to the transformation from a FT to a CFT consisting of four CFT components. The system-level component "C1" represents the risk of a system. This component contains other (sub-)CFT components that represent risks of the subsystems (i.e., the architectural components of the system). The repeated basic events "Power Unit Down" are merged to form the component "Power Unit". This solves the issue of the repeated events. The basic events representing "CPU Down" respectively belong to the component "Main Controller" and "Auxiliary Controller". The CFT component "Power Unit" inputs the failure into both the upper components via in/out ports.

Each CFT component corresponds to a system architectural component. With the help of in- and out-ports, engineers may treat each CFT component as a black box without considering the details. Components are influenced or influence others only via ports. A CFT may output data to multiple successors. Thus, the CFT is no longer a tree, but a directed acyclic graph (DAG). A CFT component may iteratively contain other components like the relations between the architectural components in the system hierarchical model. Thus, the CFT provides two types



(a) Fault tree.

(b) Component fault tree corresponding to the fault tree of (a).

Figure 2.3: Component Fault Tree (example from [108]).

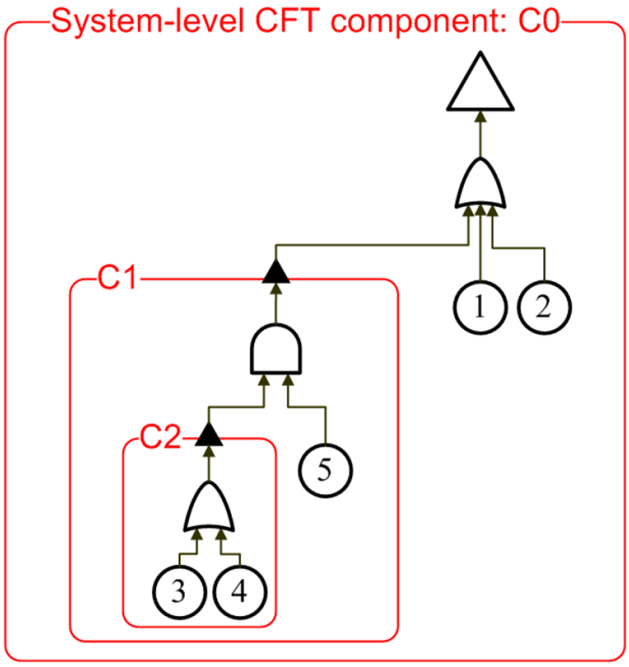


Figure 2.4: The multi-level nesting of the CFT components. Component "C2" is nested in component "C1" and "C1" is nested in the system-level CFT component "C0".

of structures: the nesting relations of CFT components and the logical structure of the CFT component (shorter form: "CFT structure" or "logical structure"), which represents the logical data flow. For example, in Figure 2.3, the system-level CFT component contains the other three components that are connected by a logical structure. Figure 2.4 demonstrates a CFT that has a multi-level nesting structure. The CFT Component "C2" is nested in component "C1" and "C1" is nested in the system level CFT component "C0".

In some cases, multiple technical components of a system may be modeled as CFT components that have the same logical structure, i.e., the same interface specification. These CFT components share the same structure (including the failure probabilities of nodes in the structure) and have different inputs and outputs. This supports the reusability of CFT components.

2.1.2.3 Failure Propagation

Failure propagation of a basic event represents the causal dependencies between failure mechanisms and the way that a failure of the basic event propagates through the system. The failure propagation is represented along a critical path that starts with an initial basic event, goes through the sequentially caused intermediate failures and ends at the top event. For the qualitative analysis, the critical path concept works on investigating how the top event is step-by-step achieved from the specific basic events. For the quantitative analysis, the critical path concept may represent the quantitative failure propagation, i.e., how basic events contribute their failure probabilities to that of the top event by logical gates. An example presented in Figure 2.5

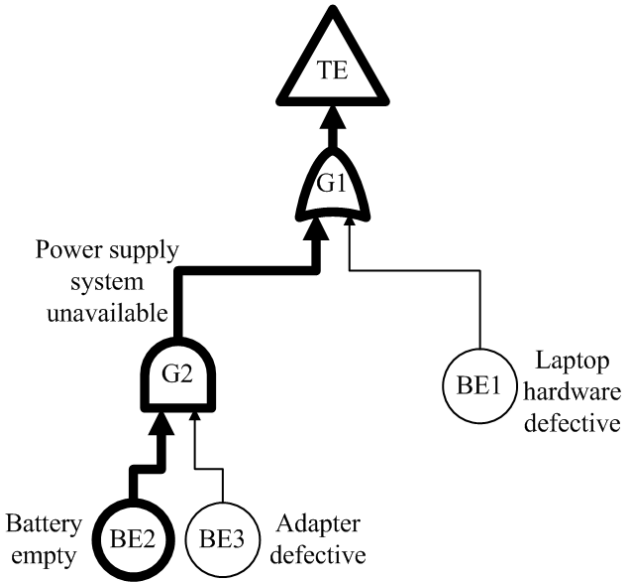


Figure 2.5: Critical path of the specific basic event "BE2". The path sequentially presents "BE2", "G2", "G1", and "TE".

shows the critical path of the specific basic event "BE2". The critical path contains the basic event "BE2", the AND-gate "G2", the OR-gate "G1", and the top event. The failure "Battery empty" (basic event "E2") may cause an intermediate failure "Power supply system unavailable" by the AND-gate "G2". Then, the intermediated failure leads to the failure of the top event "Laptop unavailable" by the OR-gate "G2".

2.1.3 Minimal Cut Sets

Minimal Cut Set (MCS) analysis [64, 67, 110, 111, 139, 199, 214] provides a method based on FTs. An MCS is a smallest combination of basic events that can together cause a top event to occur. The result of the MCS analysis consists of all possible MCSs. Each MCS can be treated as a specific scenario that can lead to the top event. An example shows MCSs of a FT having an AND-gate and an OR-gate (Figure 2.6). The occurrence of the top event is formulated as a logical representation: $(BE_1 \vee BE_2) \wedge BE_3$ (Figure 2.6 (a)). It can be transformed to $(BE_1 \wedge BE_3) \vee (BE_2 \wedge BE_3)$. Thus, there are two MCSs: $MCS_1 = \{BE_1, BE_3\}$ and $MCS_2 = \{BE_2, BE_3\}$ (Figure 2.6 (b)). The count of basic events of an MCS is called *order* of the MCS [159] or *size* of the MCS [111]. In the above example, the order of MCS_1 is 2, because it contains two different basic events. The order of an MCS is inversely proportional to its safety. An MCS with order 1 is very critical for a system, because the top event may be triggered by only one basic event (single failure).

Besides the qualitative analysis, engineers may also perform the quantitative analysis for FTs. The failure probability of an MCS is product of failure probabilities of the contained basic events (BEs):

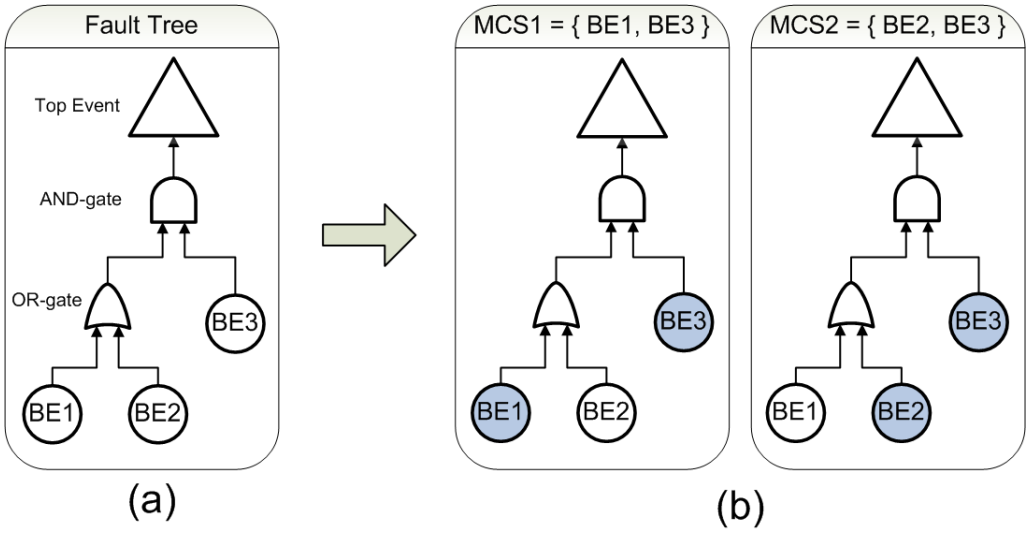


Figure 2.6: Minimal Cut Sets. (a) FT. (b) MCSs of the FT. $MCS_1 = \{BE_1, BE_3\}$ and $MCS_2 = \{BE_2, BE_3\}$. The order of MCS_1 is 2, because it contains two basic events.

$$P_{MCS} = \prod_{BEs \in MCS} P_{BE_i}$$

Failure probability of a top event is the sum of the failure probabilities of MCSs of the top event. It is formulated as

$$P_{TopEvent} = \sum_{MCSs} P_{MCS_i}$$

The *dominant cut sets* is the cut sets that contribute significantly to the top event probability [142]. The basic events of the dominant cut sets should be prioritized. A given FT will have a finite number of unique MCSs; however, the scale of MCSs may be very large [142]. An example given by [143] was that a FT with 299 basic events and 324 gates had more than 64 million MCSs. According to the rules of the CFT analysis, the MCS method is adaptable to analyzing CFTs as well.

In addition, the MCS analysis can be used to identify the critical basic events with respect to the number of occurrence. A basic event may contribute to a specific undesired system failure (top event) by means of multiple scenarios (MCSs). In this way, a basic event that appears in many MCSs can be considered as a critical basic event.

2.1.4 Importance Analysis

Importance analysis is a widely used quantitative measure based on FTs [48, 89, 93, 136, 137, 142, 159]. Vesely et al. suggested that, in general, more than 90% of the failure probability of a top event was due to less than 20% of the basic events [142].

This implies that engineers only need to focus on a small subset of basic events having major contribution. To identify those, the importance analysis estimates the respective contributions of basic events with regard to the failure probabilities of basic events and the logical relations between basic events. The importance analysis is usually used in three areas [195]:

- (Re)design of the system: adding or removing system components.
- Test and maintenance: optimizing the test and maintenance strategy.
- Daily configuration control: estimating the effect when a component fails.

As a result, the importance analysis assigns each basic event a value of importance. The widely used methods of importance analysis are Fussell-Vesely importance [68, 136, 137, 159], Birnbaum importance [25, 136, 137], Criticality measure [136,137,159], Risk Reduction Worth (RRW) [38] and Risk Achievement Worth (RAW) [136, 137].

The most commonly used Fussell-Vesely (FV) measure [68,137,159] assigns each basic event an importance value between zero and one: the larger the value, the more important the basic event. The sum of the importance of all basic events of a system may be greater than one since, in some cases, simultaneous failures of multiple sub-systems may cause the system failure [51]. With the help of MCSs, the FV importance of the basic event E can be formulated as

$$FV_E = \frac{\sum P_{MCSs\_E}}{P_T}$$

, where P_{MCSs_E} is probability of an MCS that contains basic event E, and P_T is probability of the top event T.

2.1.5 Sensitivity Analysis

Sensitivity analysis is a method for estimating the relations between changes of inputs and resulting changes of the output. This method is widely applied in various domains. In the safety analysis domain, sensitivity analysis based on FTs investigates the resulting impacts on top event caused by the changes of basic events [48,50,93,95, 136,142,149]. Sensitivity analysis usually is used for the following aspects [48,50,149]:

- Investigation of the impacts on the top event when modifying basic events.
- Identification of the safety vulnerabilities of system design.
- The appropriate improvement solutions for safety vulnerabilities.

For the first aspect, the sensitivity of basic events is usually used to analyze the accuracy of failure probabilities of basic events. For obtaining the sensitivity of basic events, engineers first change the failure probability of each basic event by the same sensitivity factor. According to the changes of the basic events one at a time, engineers separately investigate the impacts of the failure probability of the top event. Finally, engineers can identify the most sensitive basic events that cause the largest impact of the top event. In many cases, some importance measures, e.g., Birnbaum importance, can be used to analyze the sensitivity of basic events. The second and third application aspects of the sensitivity analysis are implemented by combining with the importance analysis. The consolidated process for improving safety of systems is described in section 2.1.6 next.

2.1.6 Safety Improvement Process

In order to efficiently identify the ways for improving the system safety, engineers usually carry out an iterative process combining the importance analysis and the sensitivity analysis by simulating a series of modifications of the system design according to the system vulnerabilities. This method was described in [34, 42, 48–50, 138, 159]. Contini et al. [48–50] extended the method in order to perform the parallel analysis of multiple FTs. Caputo et al. [34] applied the method in the Borexino Experiment. Murtha [138] described an application of the method in vehicle design.

The design modification commonly involves the aspect by replacing the vulnerable parts of the system by ones having a better failure performance (substitution concept) or adding identical redundant parts (redundancy concept). Finding satisfactory modifications in general is a non-trivial task underlying several constraints and restriction for which formal methods are not always available. Each iteration of the safety improvement process consists of mainly three steps (Figure 2.7):

1. Perform the importance analysis to identify the basic event having the highest contribution.
2. Find the hardware component related to that basic event and modify the system design by replacing the component by another one featuring a better quality or by introducing identical ones in order to increase redundancy (Figure 2.8).
3. Update the FT model according to the design modification and calculate the new failure probability of the top event. If the failure probability of the top event is reduced to an acceptable range, we stop the improvement process, otherwise go back to step 1 for next iteration.

   There may be alternative modifications because of multiple modification ideas aiming at the identical important basic event (Figure 2.8), and multiple important basic events (Figure 2.9). If required, engineers need to determine the optimal modification taking the effect of the modification and the constraints of modification into account, e.g., the costs of the modifications.

As a result of the safety improvement process, engineers can obtain one or more improvement solutions (Figure 2.10). Each solution consists of a series of design modifications. Choosing the proper one is a trade-off between the constraints regarding the improvement and the safety improvement that actually has been achieved. An important assumption is that basic events are stochastically independent so that the change of a basic event does not influence other basic events. An application example in section 3.2.3.1 shows an iteration of the improvement process.

2.1.6.1 Decision Criteria

To a complex system, the number of possible solutions might be very large. Usually, there are two ways to reduce the amount of solutions:

- Optimizing the modifications in the construction process of solutions. When multiple modifications are identified, engineers may determine the optimal one(s) and refuse the inappropriate alternatives.

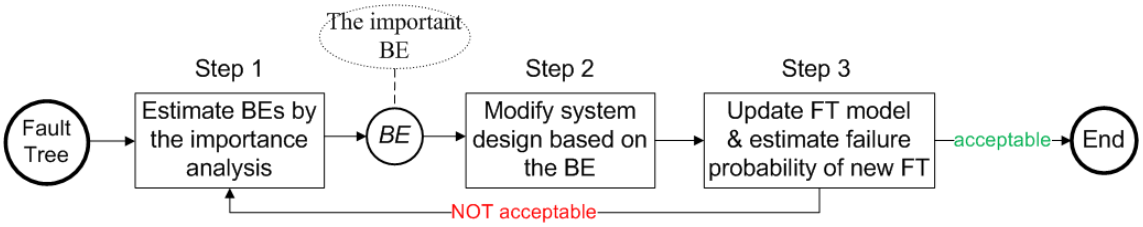


Figure 2.7: The safety improvement process. Step 1: analyze the importance of basic events and identify the most important one. Step 2: modify system design according to the most important basic events. Step 3: update the FT and calculate the new failure probability of the top event. If the new value is acceptable, stop the safety improvement process, otherwise go to step 1.

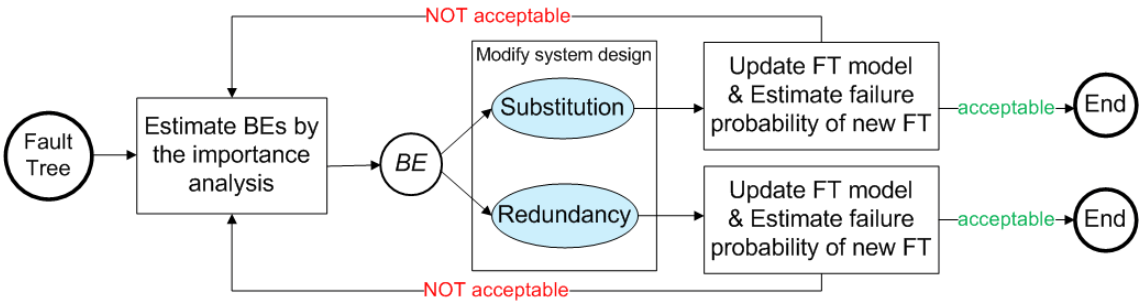


Figure 2.8: Different modification ideas of an important basic event. The commonly used modification concepts are the substitution concept and the redundancy concept.

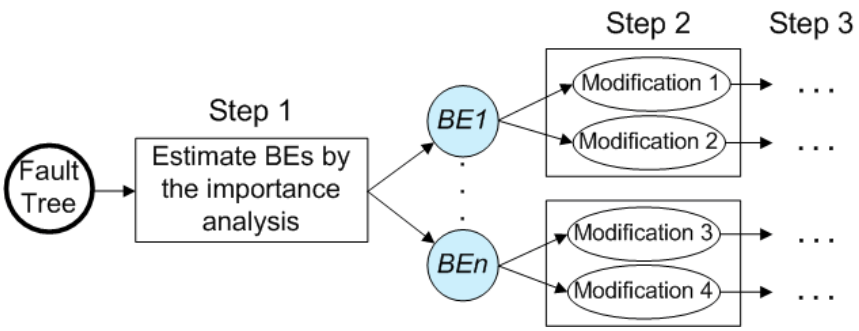


Figure 2.9: Multiple important basic events. More than one basic event has an importance similar to the largest importance value.

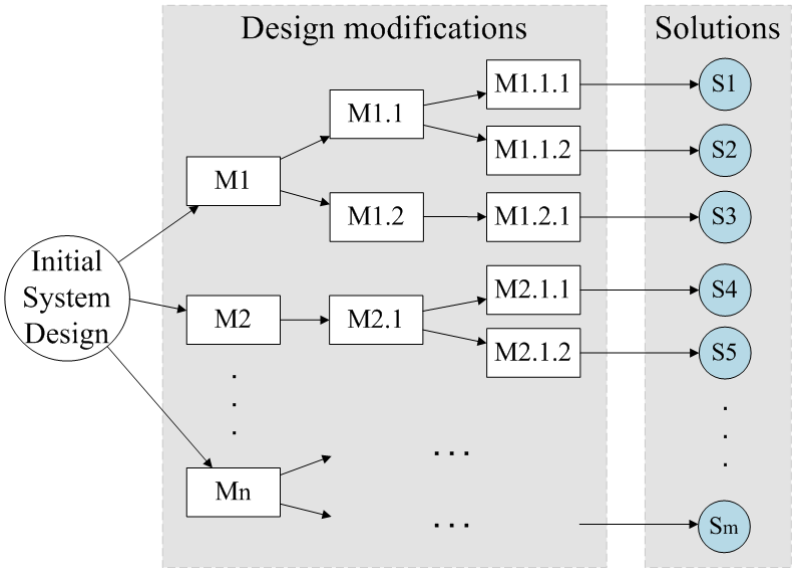


Figure 2.10: Multiple solutions. The alternative design modifications may constitute multiple improvement solutions.

- Reducing the constructed solutions. When multiple available solutions exist, engineers may identify the optimal one(s).

The objective of the optimization of solutions determines the criterion of modifications. The commonly used criteria are described as follows:

- Identify the improvement solution consisting of the fewest modifications. In some cases, in order to keep the system stable in the improvement process, engineers want to modify as few components as possible. In this case, a solution needs to be found with fewest modification steps. A modification that causes the most reduction of the failure probability of the top event should be treated as the optimal one.
- Identify the most cost-effective improvement solution. This criterion takes the cost of modifications into account. To efficiently use the resource, the most cost-effective modification should be considered as the optimal one.

2.2 Representation Concepts of Safety Analysis

2.2.1 Representation Concepts of the CFT

2.2.1.1 Single CFT

Although documents standardize the graphical symbols of the FTs [91, 142] (Figure 9.1), there are still additional elements for CFTs: CFT component and ports (Figure 2.3). The CFT components are represented as rectangles in the logical structures of the parent CFT components. The in- and out-ports are represented as small triangles. The tool ESSaRel [193] implements the CFT analysis using the standard graphical symbols defined in [108].

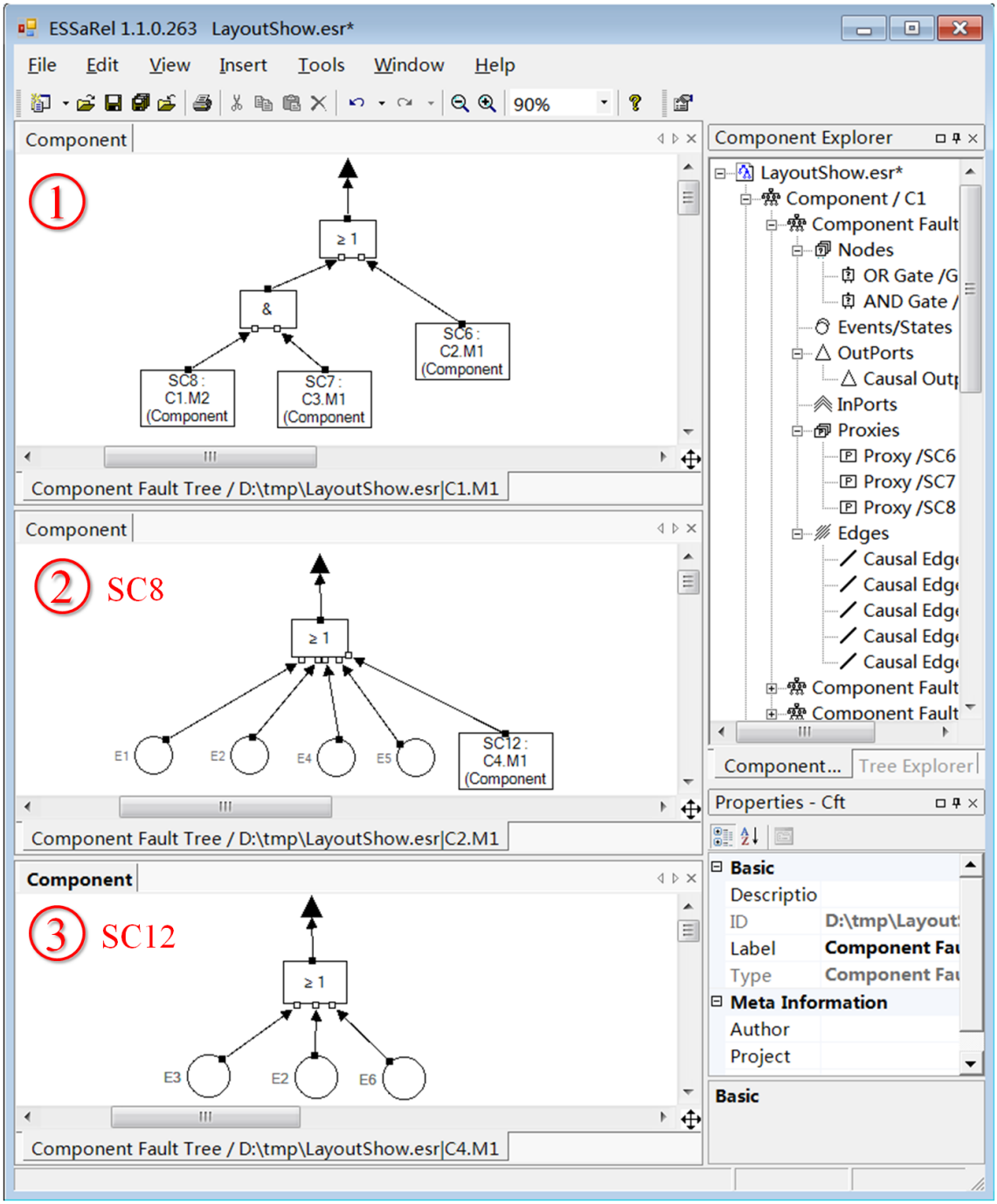


Figure 2.11: Example of the representation method of CFTs (produced using ESSaRel [193]). The structure of the system-level CFT (view (1)) and those of the sub-CFT components (view (2) and view (3)) are shown in separate views.

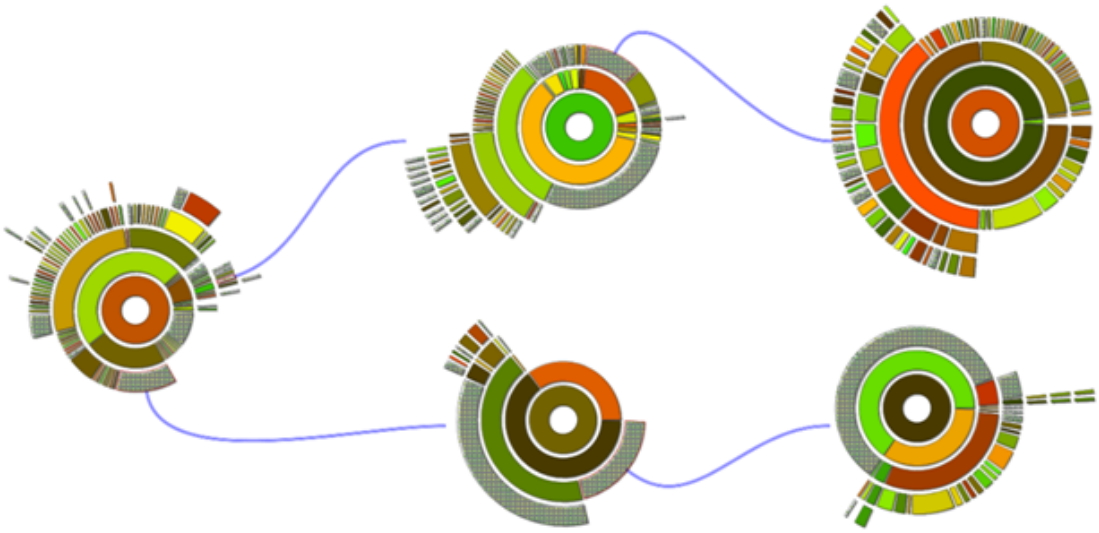


Figure 2.12: Visualization Concept of Fault Forest (produced by [26]). Sunburst layout represents the single CFT. The sunbursts are connected by curve lines to form a fault forest.

The logical structures of the (sub-)CFT components are presented over separate views (Figure 2.11). When double-clicking a rectangular symbol representing a sub-CFT component, the structure of this sub-CFT component will be shown in an individual view. The views in Figure 2.11 appear in the following way: double-clicking the leftmost rectangular symbol representing the sub-CFT component "SC8" in view (1), view (2) appears and show the structure of the sub-CFT component "SC8"; then double-clicking the rectangular symbol representing the sub-sub-CFT component "SC12" in view (2), view(3) appears and presents the structure of the component "SC12".

2.2.1.2 Multiple CFTs

Logical connections of multiple CFTs forms a network structure called a Fault Forest. A visualization concept of fault forest was proposed in [26] (Figure 2.12). The concept uses a sunburst diagram to represent the single CFT and connects the sunbursts with curve edges. In a sunburst view, the central ring represents the top event and the segments represent the internal nodes of the logical structure of CFT. The logical connections between a node and the child nodes in the CFT are represented as a segment and the child segments. Colors are used to encode the failure probability of the CFT nodes. With the help of this visualization, engineers may analyze the individual CFTs and the logical connections of CFTs with respect to the multiple top events.

2.2.2 Representation Concepts of MCS

The currently applied representation concepts of MCSs concentrate on text or tabular forms.

2.2.2.1 Plain Text

The textual forms list MCSs and their basic events with basic information in plain text. Advanced functions, e.g., sort, depend on the text editors. The CFT analysis tool *ESSaREL* [193] presents MCSs in plain text (Figure 2.13). In each paragraph, it shows an MCS with its ID, and the IDs and labels of the included basic events. It also shows the general information of the current analysis, such as an ID of the top event and count of the MCSs. In some cases, failure probabilities and the orders of MCSs may be presented optionally in plain text [162].

2.2.2.2 Data Table

Data tables are able to more clearly represent the information of the MCSs than through using plain text, and provide possibilities for data manipulations [6,100,101] (Figure 2.14). Data tables present MCSs in rows and the associated information in columns. Besides the IDs of MCSs, the tables usually also show probability, contribution, and the order of MCSs in columns while the possibly used basic events are listed in subsequent columns. Engineers may manipulate data in data tables with few additional visualization properties, e.g., sorting of MCSs. Additionally, MCSs may be truncated regarding the order or the failure probability in order to filter out the unimportant MCSs. The data table concepts provided by most of the FTA tools are quite similar.

Some tools use separate tables to show the MCSs and their basic events, e.g., Faulttree+ [100]. When selecting an MCS from a data table that summarizes all MCSs, the basic events included by the MCS are displayed in another table that shows detailed data of each basic events (Figure 2.15). Additionally, the general information associated with MCSs may be presented outside the table, e.g., the name of the FTA, probability of the top event and statistical information of MCSs.

2.2.2.3 Critical Path Highlighting

Besides the representation concepts that aggregate the information of MCSs, a representation concept regarding the critical path in coordination with MCSs is also provided [29, 100, 161, 186]. The concept does not compactly integrate the overview of MCSs with the FT structure, but rather provides an interactive coordination. When selecting an MCS from the data table, the critical paths of the basic events included by the MCS are highlighted by colored border (Figure 2.16).

2.2.2.4 3D Visualization Concept

A three-dimensional visualization concept of the MCS analysis, called *CakES*, was proposed in [4, 5] (Figure 2.17). This visualization worked on integrating the 3D CAD models of a system with the MCS analysis. This shows the corresponding relations between the 3D models of the physical parts and the MCSs according to the commonly related basic events. The MCSs are represented as cylinders and arranged in a 3D circle layout in three different hierarchies with respected to the criticality of MCSs. Colors are assigned to the cylinders based on the failure probabilities of MCSs. When analyzing an MCS, the corresponding cylinder rises from its hierarchy,

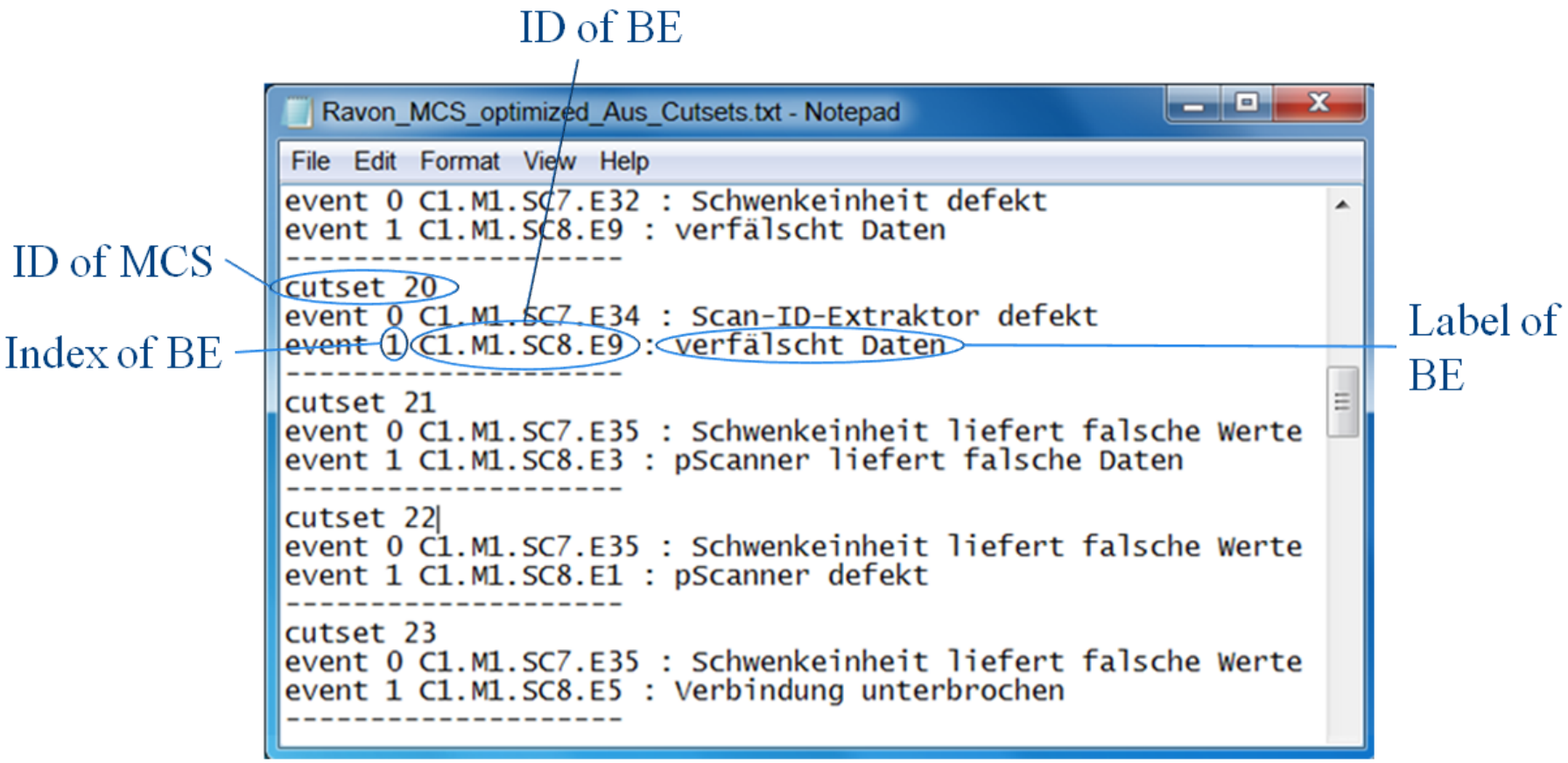
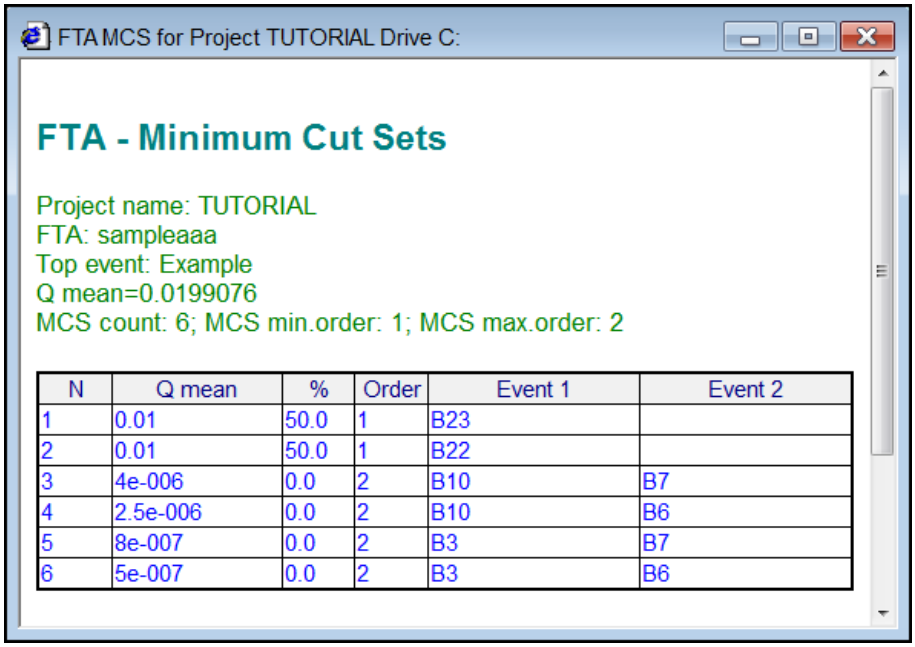


Figure 2.13: Plain text for the MCSs (data generated using ESSaREL [193]). Each paragraph separated by a dashed line presents data of an MCS: the ID of the MCS, the IDs and labels of the basic events.



N	Q mean	%	Order	Event 1	Event 2
1	0.01	50.0	1	B23	
2	0.01	50.0	1	B22	
3	4e-006	0.0	2	B10	B7
4	2.5e-006	0.0	2	B10	B6
5	8e-007	0.0	2	B3	B7
6	5e-007	0.0	2	B3	B6

Figure 2.14: Data table representation for MCSs - 1 (produced using RAMCommander [6]). MCSs are listed in rows and sorted by the failure probability (the column "Q mean"). The associated data as well as the basic events are presented in columns.

and shows the 3D models of the physical parts related to the basic events of the MCS. The commonly used interactions for 3D views, e.g., rotating, are also provided for this concept.

2.2.3 Representation Concepts of the Importance Analyses

2.2.3.1 Data Table

The result of the importance analysis of the FTA is a set of importance values of basic events. The result is usually summarized and represented using a data table by most of the FTA tools [6,48,50,100,161,186]. The basic events are represented in rows and the importance values as well as the basic information of basic events, e.g., the failure probability, are represented in columns. For each basic event, the data tables may show the results of multiple importance measures in respective columns.

2.2.3.2 Graphical Concepts

For any importance measure, the result is a typical 2-dimensional array data. The graphical forms are suitable to represent the data-aggregated result [6, 48, 50, 162, 186]. The most commonly used forms are the histogram and the pie chart (Figure 2.19). Besides these forms, there are still commonly used alternative graphical concepts to represent the result of the importance analysis either in 2D or in 3D forms (Figure 2.20), such as the scatter plot, and the area chart. BlockSim [162] (Figure 2.21) assigns colors to the histogram according to the failure probability of basic events. Additionally, BlockSim proposed a variant of pie chart called "square pie

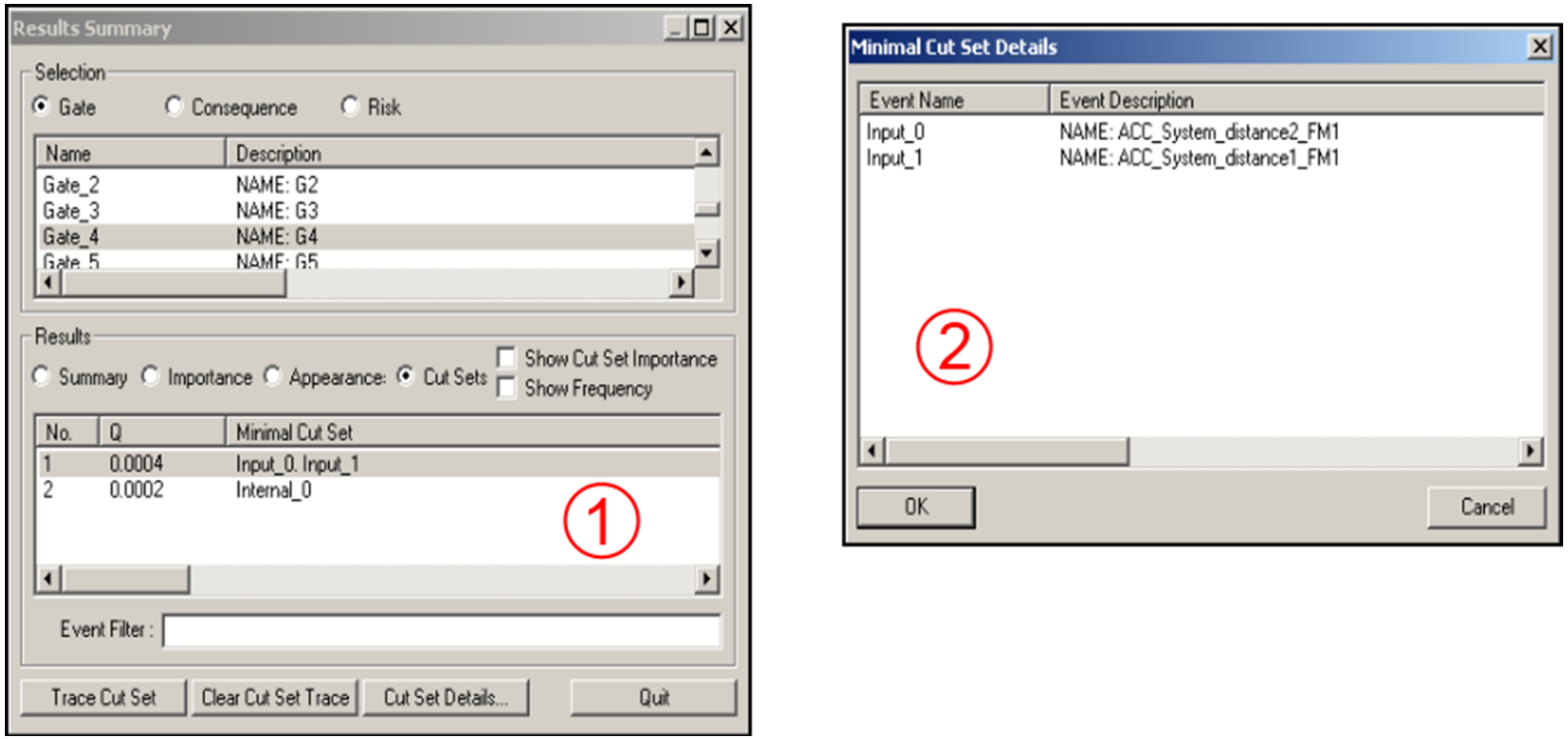


Figure 2.15: Data table representation for MCSs - 2 (produced using Faulttree+ [100]). View (1) presents the MCSs and their IDs and failure probabilities. View (2) presents the information of basic events included in the first MCS that is selected in view 1.

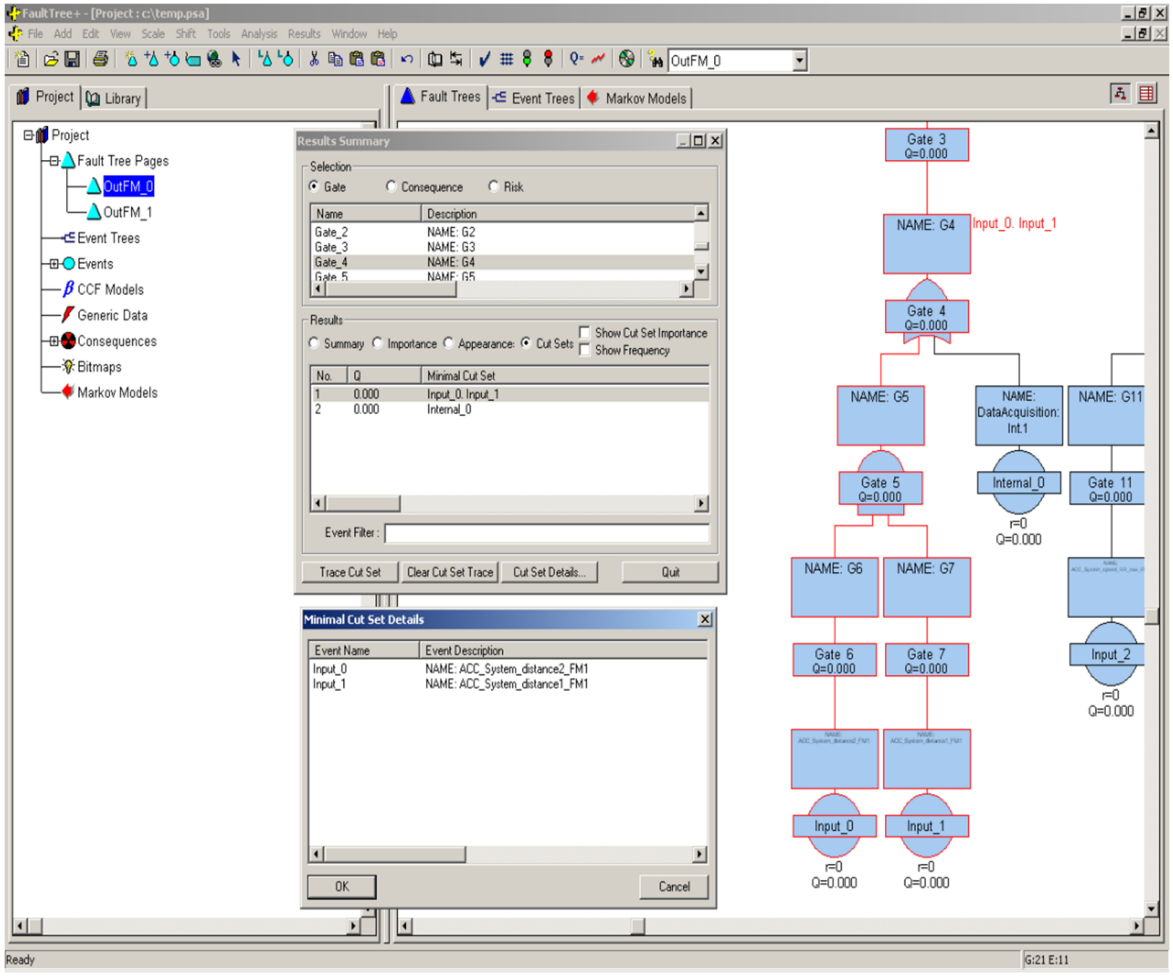
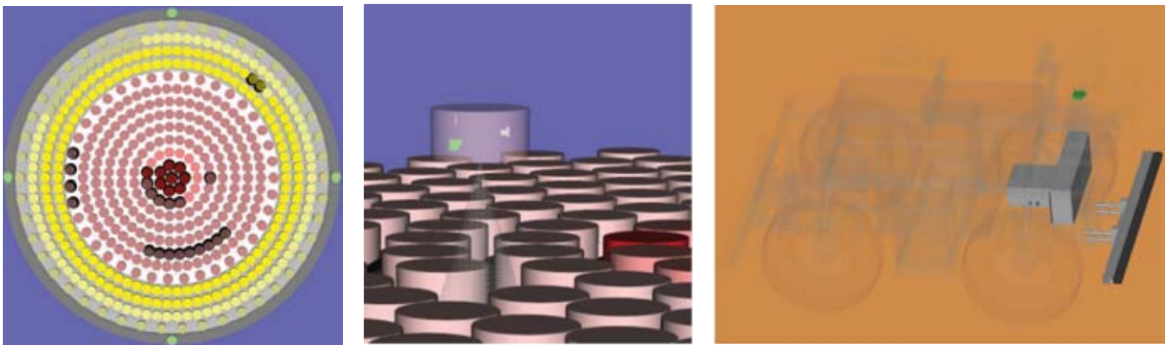


Figure 2.16: Critical path highlighting. The critical paths corresponding to the selected MCS are highlighted by red borders (produced using Faulttree+ [100]).



(a) (b) (c)

Figure 2.17: The 3D representation of MCSs: CaKES (produced by [5]). MCSs are represented as 3D cylinders. Cylinders are hierarchically arranged in a circle layout and assigned with colors according to the failure probabilities of MCSs. (a) The view of MCSs. (b) The view of 3D models of physical parts related to the specific MCS. (c) The overall 3D model of a robot.

FTA Importance/Sensitivity for Project TUTORIAL Drive C:

FTA - Importance & Sensitivity Analysis

Project name: TUTORIAL
FTA: sampleaaa
Top event: Example
Q mean=0.257558

N	Code	Occurrenc	Q me	FV Imp.	FC	RDF	RIF	Sens. high	Sens. low	Sensi- tivity
1	B22	1	0.25	0.970655	0.960874	25.5584	3.88262	1	0.0348253	28.7148
2	B23	1	0.01	0.0388262	0.0291174	1.02999	3.88262	0.325053	0.250808	1.29602
3	B10	2	0.00	0.000252	0.000187	1.00019	1.36305	0.257992	0.257514	1.00186
4	B7	2	0.08	0.000186	0.000138	1.00014	1.00159	0.257879	0.257526	1.00137
5	B6	2	0.05	0.000116	8.65e-005	1.00009	1.00164	0.257758	0.257538	1.00086
6	B3	2	0.00	5.05e-005	3.75e-005	1.00004	1.36318	0.257645	0.257549	1.00037

Notes:
Occurrence - number of occurrences of the basic event in all minimal cut sets
FV Importance - Fussell-Vesely Importance (FV = Q of MCS which contains the basic event / Q of all MCS)
FC - Fractional Contribution of Basic Event (1-1/RDF)
RDF - Risk Decrease Factor
RIF - Risk Increase Factor
Sensitivity - Sensitivity Value, calculated with sensitivity factor=10

Figure 2.18: Data table representation for the importance analysis (produced using RAMCommander [6]). The table presents the importance values of basic events with respect to different important measures.

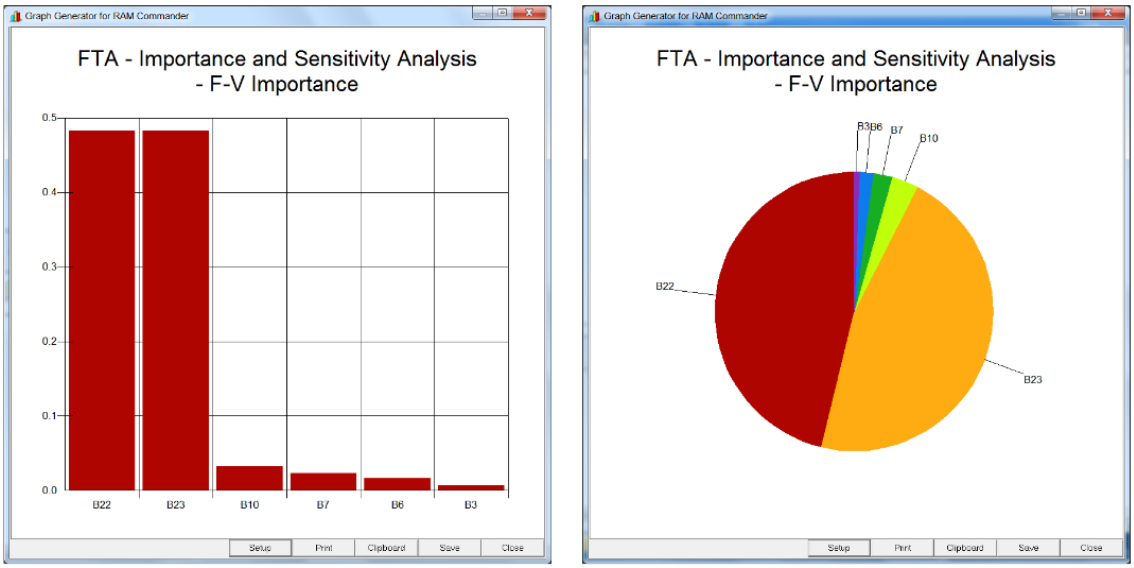


Figure 2.19: Graphical representations for the importance analysis (produced using RAMCommander [6]). Using the histogram, the importance of basic events can be assessed according to the length of the bars. The pie chart graphically represents the relative magnitude of the importance of basic events.

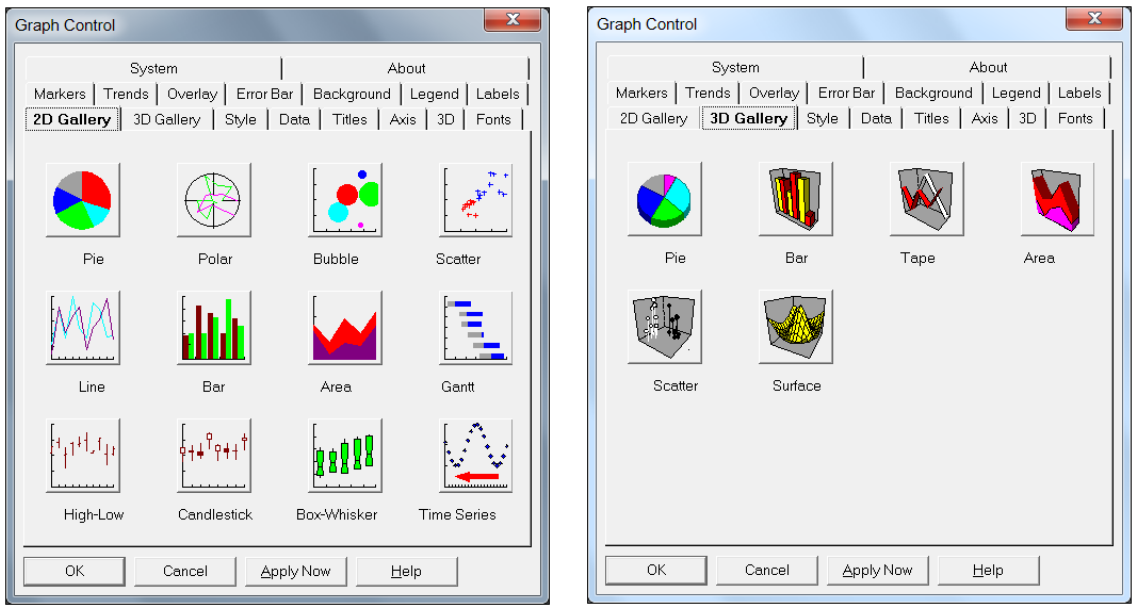
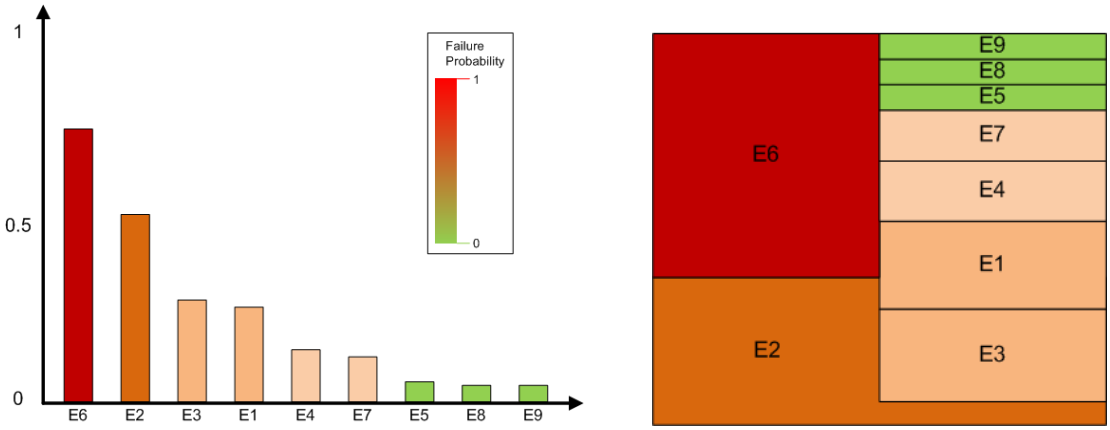


Figure 2.20: Alternative graphical representations for importance of basic events (produced using RAMCommander [6]).



(a) Histogram with colors. (b) Square pie chart.

Figure 2.21: Colored histogram and the square pie chart (concepts from BlockSim [162]).

chart" that anticlockwise arranges the basic events in descending order with respect to the importance values of basic events.

2.2.4 Representation Concepts of the Safety Improvement Process

The safety improvement process consolidates the importance analysis and the sensitivity analysis for identifying the improvement solutions of the system safety. During the safety improvement process, engineers require data from various aspects, e.g., the data of design modifications, the importance of basic events, and the FT diagram. Besides FT structures, most FTA tools usually provide separate views for the data. Data relevant to the importance analysis may be represented by various forms that are introduced in section 2.2.3. Data associated with the design modifications and the risk reduction are usually represented using data tables, e.g., the failure probability of the modified basic events, new failure probability of the updated top event, types of modifications, cost of modifications. There are few visually and interactively associations among the views. In each step of the safety improvement process, engineers need to manually access the data from the views according to the requirements.

Project CISA [48, 50] arranged data of the design modifications in the separate views using the ordinary representations. Besides the FT view and the importance analysis, most data was textually presented in data tables. An indented decision tree was used to represent the summary of improvement solutions (Figure 2.23). A design modification was represented as a square where the letter represents the type of the modification: R represented redundancy concept and E represented component substitution. The label next each square showed the ID of the related basic event as well as the ID of the design modification (in the bracket). The separate data views were logically linked to the decision tree. When selecting a node from the tree, the data associated with the node was shown in additional views.

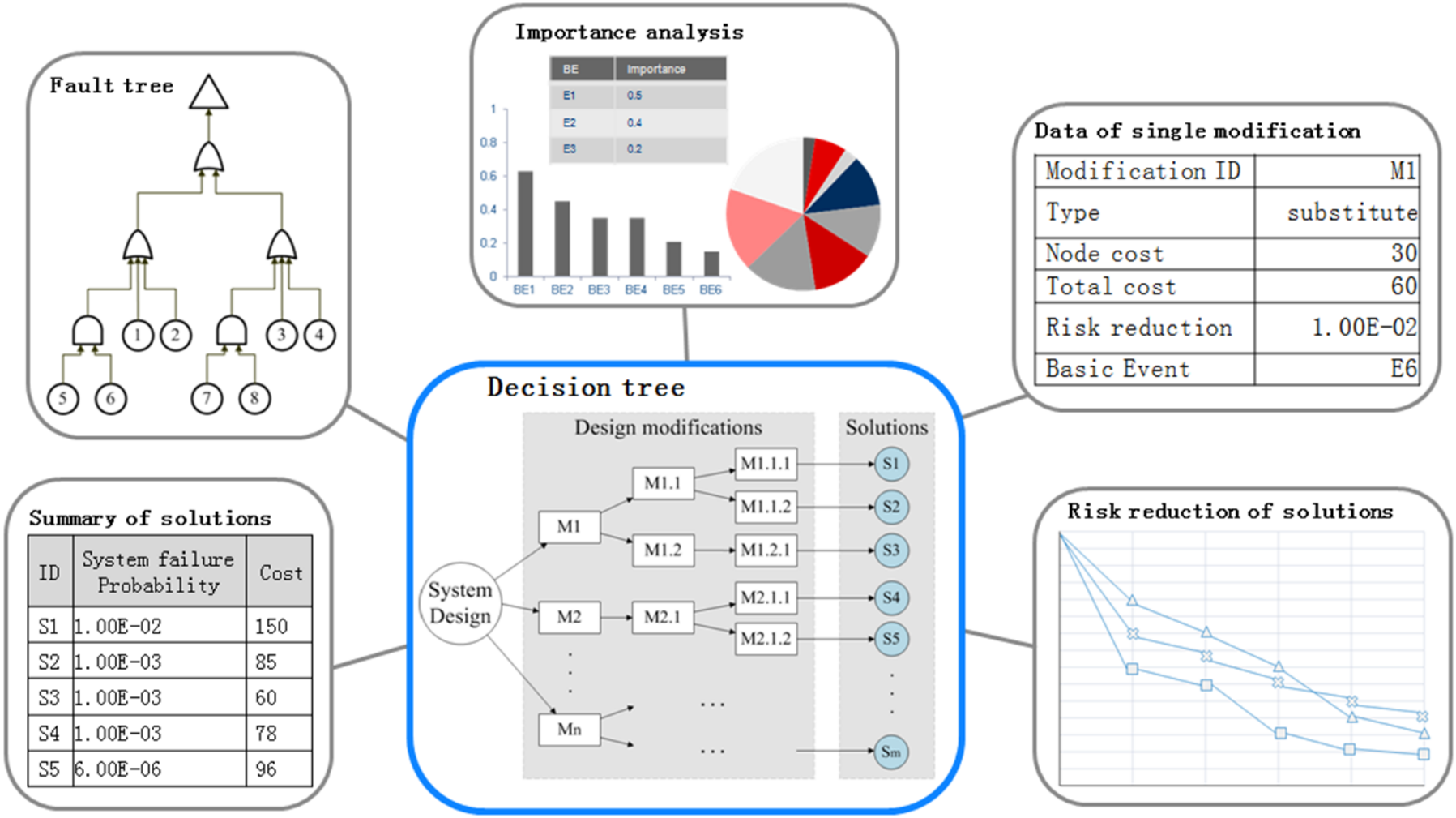


Figure 2.22: The ordinary representation concepts of the safety improvement process. Design modifications are arranged by a decision tree [50]. The data relevant to the modifications are separately presented over several views [6, 48, 50, 186]. The representations commonly used in the safety improvement process are FTs, charts or tables for importance of basic events, tables of possible solutions, tables for design modifications, and the plot for risk reduction of system.

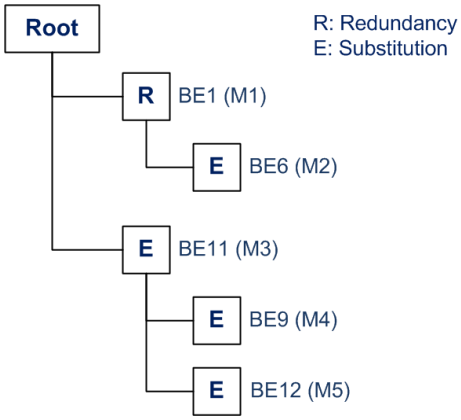


Figure 2.23: The indented decision tree arranging the improvement solutions (concept from [50]). Squares represent design modifications. The letters inside the squares represent types of the modifications. The label next each square shows the ID of the related basic event as well as the ID of the design modification (in the bracket).

2.2.5 Other Representation Concepts of Safety Analysis

2.2.5.1 Risk Matrix

In order to qualitatively estimate risk, the risk matrix was proposed and defined in [97, 141, 147]. The frequency of occurrence and severity of harm are classified into different levels (Figure 2.24). Colors are assigned to the levels in order to encode the criticality of risk.

2.3 Information Visualization

This section, the fundamentals of information visualization are introduced. The introduction is guided following the lecture of "Topics in Information Visualization and Visual Analytics" conducted by Dr. Dirk Zeckzer at the University of Kaiserslautern [210] and the book "Readings in information visualization: using vision to think" [35].

2.3.1 Graph Drawing

Graph is defined in mathematics as a pair $G = (V, E)$, where V is a finite set of vertices and E is a finite set of edges. Each edge $e \in E$ connects a pair of vertices $(v_1, v_2) \in V$ for representing the relation in between. If the edges have directions, the graph is called a *directed graph*, else an *undirected graph*. When a directed graph has no cycles, the graph is called a *directed acyclic graph (DAG)*. A *tree* is a connected acyclic graph. A graph is called *weighted graph* when edges have weights that express the strength of relations between vertices. A graph is called a *bipartite graph* when there are only two disjoint sets of vertices.

Graph drawing is a research area that focuses on the algorithms of graph that usually work on the layouts of node-link diagrams taking readability of graphs into account [191]. The commonly used layouts are:

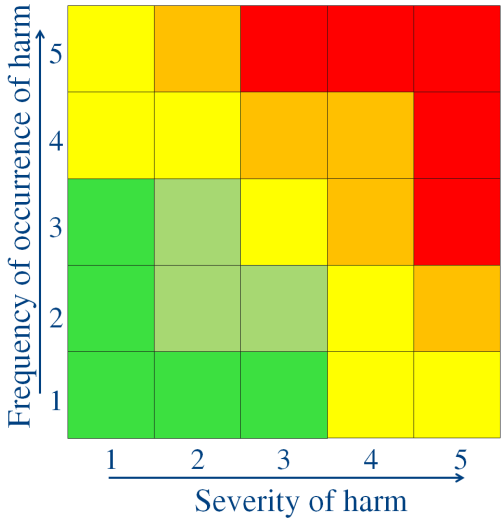


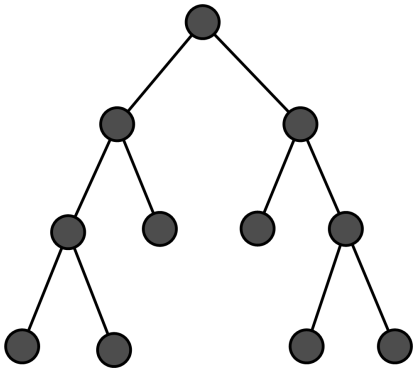
Figure 2.24: Risk matrix. Frequency of occurrence and severity of harm are respectively classified into different levels. Colors encode the levels.

- *Reingold-Tilford layout* [160, 203] (Figure 2.25 (a)) is the commonly used algorithm for drawing rooted-trees. This algorithm represents a tree by layering the nodes according to their parent-child relations in a specific direction, e.g., top-down orientation. This clearly reflects the hierarchy of data.
- *Radial layout* [58, 59, 85] (Figure 2.25 (b)) uses a polar coordinate instead of the Cartesian coordinate to represent trees. The root node of a tree is placed in the center of the radial plot and hierarchies are represented by radius. Nodes are placed on the concentric circles according to their hierarchies.
- *Balloon tree layout* [127, 135] (Figure 2.25 (c)) is a particular radial tree that places nodes around their parent node rather than the concentric circles.
- *Force-directed layout* [60, 65, 109, 192] (Figure 2.25 (d)) is also called the spring layout. The graph is treated as a physics model. The distance between nodes depends on the calculated attraction force and the repulsive force.
- *Circular layout* [57] (Figure 2.25 (e)) places graph nodes on a circle. This effectively reduces the node overlapping.
- *Hierarchical graph layout* [15, 53, 183] (Figure 2.25 (f)) represents directed acyclic graphs (DAGs) taking the hierarchy of graphs into account. This layer can be determined from either structural properties or a specified attribute of nodes.

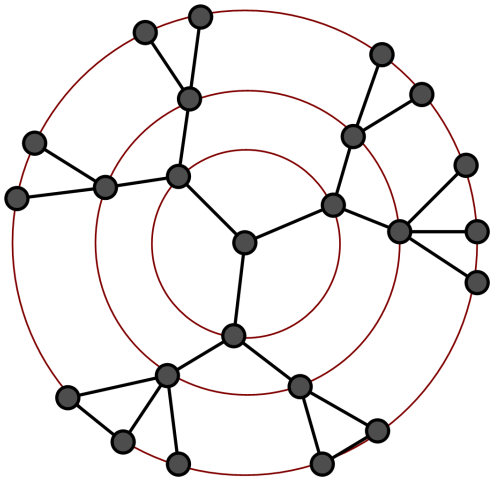
2.3.2 Visualization

Visualization is a process that represents relations or data as a visible form. This is an important branch of the computer graphics. Visualizations can be primarily classified into two types:

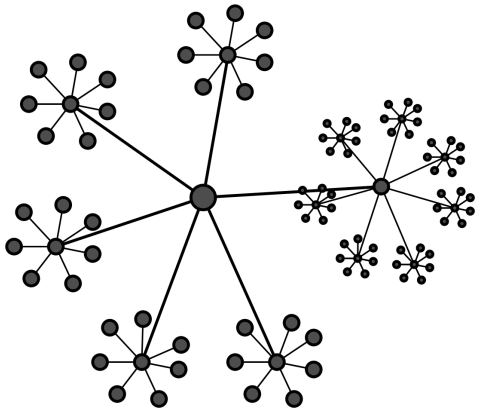
- Scientific visualization: visualizing spatial data.



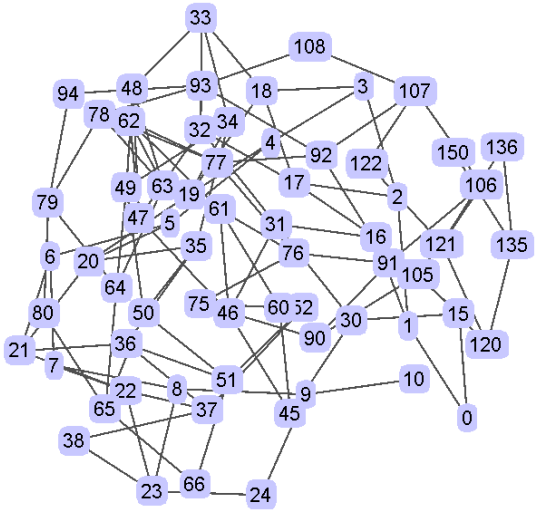
(a) Reingold-Tilford layout.



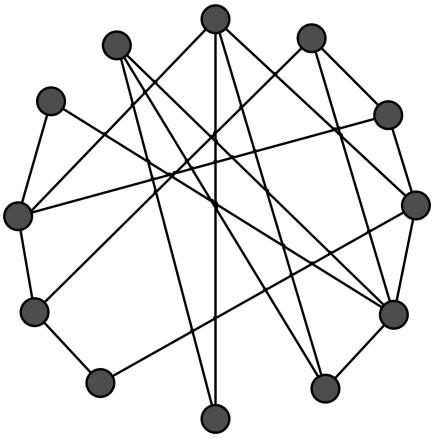
(b) Radial layout.



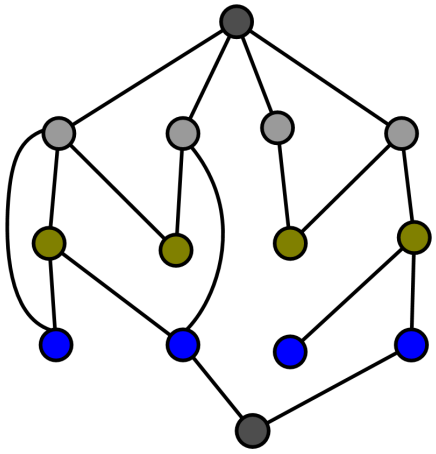
(c) Balloon layout.



(d) Force-directed layout (produced by Prefuse [77]).



(e) Circular layout



(f) Hierarchical graph layout.

Figure 2.25: Graph layouts.

- Information visualization: visualizing abstract (non-spatial) data.

Visualization is widely applied in science, engineering, and education fields. This facilitates the data exploration and improves the understanding of concepts and processes in visual ways. This also supports to intuitively represent the hidden information.

2.3.3 Information Visualization

Information visualization was defined as follows:

- "*The use of computer-supported, interactive, visual representations of abstract data to amplify cognition*" [35].
- "*Information visualization utilizes computer graphics and interaction to assist humans in solving problems*" [156].
- "*Information visualization is a set of technologies that use visual computing to amplify human cognition with abstract information*" [148].

Information visualization is typically applied in the following fields [210]:

- Statistics: visualizing the statistical data for discovering patterns.
- Social network: visualizing social relations between people.
- File system: visualizing the structure of the computer file systems.
- Software: visualizing the software architecture, evolution, and runtime data.
- Biology: visualizing the gene sequence.
- Schedule: visualizing work schedule, and train and airplane scheduling.
- Data Mining: visualizing the results of data-mining process with respect to clustering, classification, and association of data.

Information visualization facilitates analysis work with the benefits [210]:

- Intuitive data representation.
- Interactive data exploration.
- Effective pattern analysis and relation discovering in large datasets.

2.3.4 Reference model of Information Visualization

Chi et al. proposed the reference model of information visualization [35, 40, 41, 163] (Figure 2.26). This model describes the process that transforms the raw data into the graphical representations.

2.3.4.1 Data Transformation

Raw data is the unprocessed dataset collected from the source. The raw data cannot be used directly for visualization because errors or missing values may exist. Thus, it needs to transform the raw data into the case-by-variables table that is the relational description of the data. The case-by-variables table is based on the mathematical relation consisting of a set of tuples [35]:

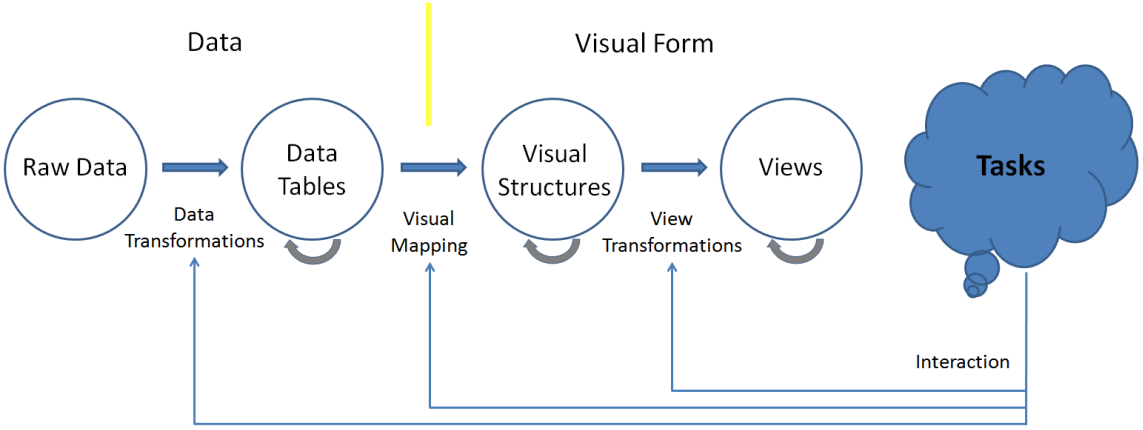


Figure 2.26: Reference model of information visualization (concepts from [35, 210]).

Student ID	N	1	2	3	4	5	6
Age	Q	26	24	22	23	24	25
Sex	N	M	M	F	F	M	F
Major	N	Math.	CS.	CS.	Math.	Phy.	Math.

Table 2.1: Example of the case-by-variables table. The rows are values of variables and the columns represent cases. The first column presents the variables of the cases. The second column represents the types of the variables.

$$\{\langle Value_{ix}, Value_{iy}, ...\rangle, \langle Value_{jx}, Value_{jy}, ...\rangle, ...\}.$$

A case is represented as a tuple that consists of the values of the variables of the case. The variables have three basic types: *Nominal* (N), *Ordinal* (O), and *Quantitative* (Q). A particular type of variable is *Link* that describes the relationship among cases. These types are the metadata and may be added to the table. The data transformation may lose or gain information from the raw data according to the analysis. For example, the statistical data can be derived from the raw data and added to the table. Figure 2.1 shows an example of a case-by-variables table that describes the dataset "Students". Each student is a case that has four variables: ID, Age, Sex, and Major. A case is formulated as:

$$Case_i = \langle ID_i, Age_i, Sex_i, Major_i \rangle.$$

2.3.4.2 Visual Mapping

Then, the case-by-variables table is mapped into the visual structure that graphically represents the data using spatial substrates, marks, and graphical properties. Determining the spatial substrate is the first step of the visual mapping. A spatial substrate is represented as a visualization layout that uses spatial positions to encode the variables, e.g., matrix view, node-link diagram. Card et al. [35] concluded the use of space in four ways:

- 1D, 2D, 3D.
- Multiple dimensions (> 3).

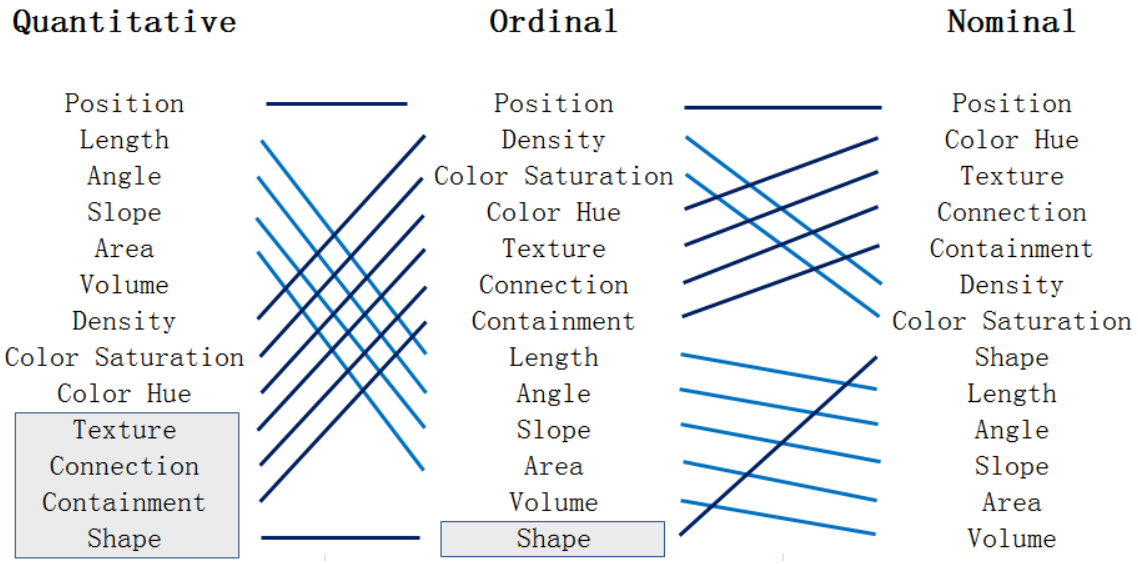


Figure 2.27: Ranking of marks and graphical properties (concept from [132]). The propertied in the gray rectangles are not suitable for representing this type of data.

- Trees.
- Networks.

The dimensions of data depends on the number of its attributes. The visual structure of one-dimensional data are usually embedded in 2D or 3D layouts. The two-dimensional data are commonly represented using a structure having two axes, e.g., scatter plot. The three-dimensional data may be represented either in a 3D space or by combining a 2D layout with an additional graphical property, e.g., color. The multiple dimensional data are similar to 3D data, which are represented by integrating 2D/3D layout with use of graphical properties or by combination of layouts. Trees and networks represent the relations among data objects. Trees represent the hierarchical structure, i.e. the parent-child relationship. There are two basic representation concepts for trees: node-link diagram and space-filling diagram. Networks are commonly represented by node-link diagrams. The matrix-based layout is an alternative representation for networks. The representations of trees and networks are introduced in section 2.4 regarding the previously proposed visualizations.

A spatial substrate only represents limited variables. It still needs to apply marks and graphical properties to the visual structure. The commonly used marks and graphical properties are [35]:

- marks: points, lines, areas, and volumes;
- graphical properties: connection, enclosure, position, length, angle, slope, density, color saturation, color hue, texture, shape.

Mackinlay [35, 132] proposed a ranking for the marks and graphical properties with respect to quantitative data, ordinal data, and nominal data (Figure 2.27). This is a top-to-bottom ranking, e.g., the property "position" is preferred for all three kinds of data.

2.3.4.3 View Transformation

The view transformation works on dynamically exploring the visually encoded data using interactions that involve three main aspects [35]: use of locations, viewpoint control, and distortion of visual structures. The commonly used interactions are introduced in section 2.3.5.

2.3.5 Interactions

2.3.5.1 Direct Manipulation

Shneiderman introduced this interaction in [168]. The user is allowed to directly select, move, rotate and resize a visual object. In order to identify the selection, the selected objects are usually highlighted. This concept provides easily understandable operations and rapid visual feedback.

2.3.5.2 Dynamic Queries

Dynamic queries [3] allow the user to explore the visualized data by dynamically adjusting query conditions using interfaces. Usually, the user interactively modifies parameters of the visualization by changing sliders or buttons for different query conditions. This technique instantly responds to the query and graphically represents the results.

2.3.5.3 Panning

Panning is an interaction that allows the user to smoothly move the display space of a large image in order to show the particular part of the image that is currently out of the screen. Usually, the user is allowed to directly drag the display space until the desired part is visible.

2.3.5.4 Zooming

(Geometric) Zooming allows to enlarge or shrink the display space of an image while maintaining the size of the image. Zooming supports the user to dynamically switch views between the overview of a large image and the specific part of the image.

2.3.5.5 Semantic Zooming

Semantic zooming displays the detailed semantic content inside the (zoomed) geometric shape of a visual object (Figure 2.30). This concept supports to view more semantic data of the visual object rather than the detailed geometric information. Summers et al. [184, 185] proposed the continuous semantic zooming for the relational structures. This allows to view the detailed semantic information of the specific visual object while maintaining the relational structures among the objects as the context. Semantic zooming usually cooperates with the Focus+Context concept.

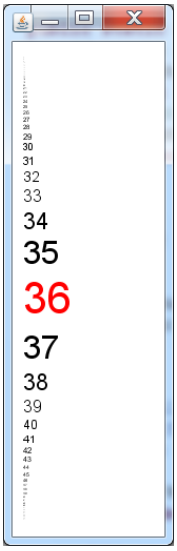


Figure 2.28: Fish-eye menu (produced using Prefuse [77]). The most interesting number "36" has the largest size. The size of other numbers depend on the distance to the number "36". The larger the distance is, the smaller the size is.

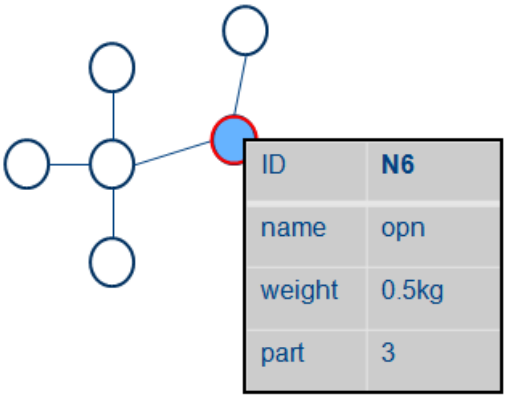


Figure 2.29: Details-on-Demand concept. The pop-up view is displayed only when being requested for presenting the details of the desired node.

2.3.5.6 Details-on-Demand

Details-on-Demand [170] is a technique for the balance between overview and details. This allows the details of visual objects to be hidden until the user desires to view (Figure 2.29). The technique is usually implemented as follows: select an object in an overview and then show the detailed information of this object using a pop-up view. Details-on-Demand concept efficiently uses the limited screen space to represent large datasets by reducing the amount of the detailed information.

2.3.5.7 Degree-of-Interest (DOI) Distortion

Degree-of-Interest (DOI) distortion was proposed in [66]. The DOI distortion presents more details for the important items and less details for the unimportant items. It needs to define the level of detail (LOD) and the DOI function. The LOD describes how much detail of an item needs to be shown. The DOI function determines how a visible item is rendered according to its LOD. A famous application is the fish-eye technique [66] (Figure 2.28).

2.3.5.8 Focus+Context

Focus+Context technique that applies the DOI concept is widely used in human-computer interaction design [1, 19, 20, 35]. Focus+Context allows to dynamically zoom the focused content while maintaining the context information (Figure 2.30). The most commonly used application is that the user investigates the detailed data while maintaining the overview as the context for navigation.

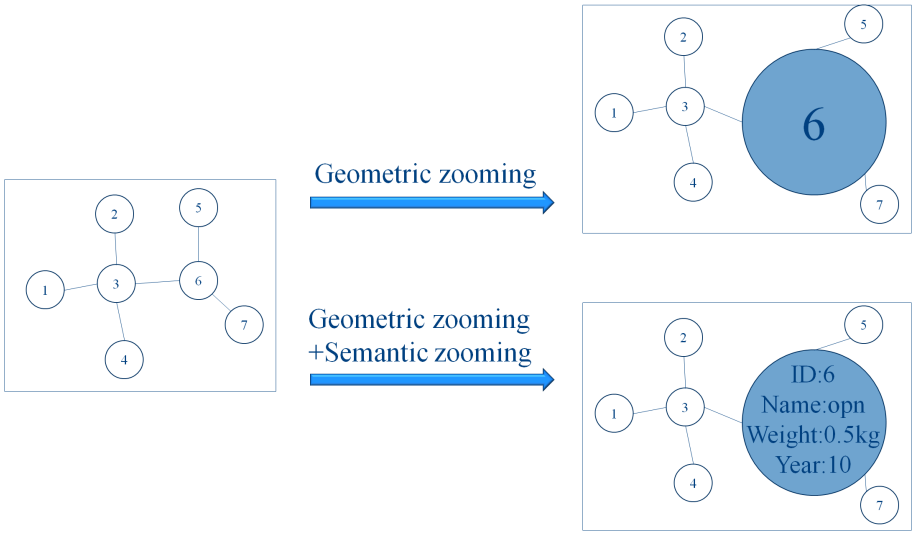


Figure 2.30: Focus+Context concept. The focused object is geometrically zoomed in. By combining with semantic zooming concept, the semantic content of the focused object is shown.

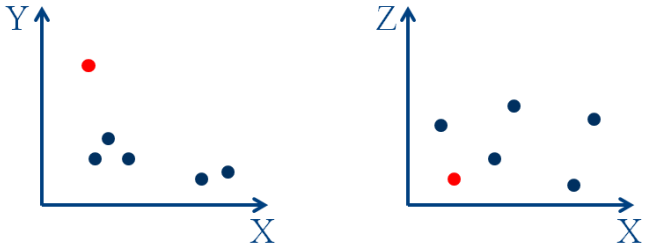


Figure 2.31: Brush-and-Linking concept. There are two scatter plots for representing the identical three-dimensional dataset. The red nodes represents the same object in two plots.

2.3.5.9 Hyperbolic Space

Hyperbolic space technique [122] maps visual structures from euclidean space to hyperbolic space that is represented as a sphere. In the center of the sphere, the visual objects keep large size. The larger the distance of an object from center of the sphere is, the smaller the size of the object has. The hyperbolic space is infinitely large at the edge of the sphere. When selecting the interesting object, the object will be moved to the center of the sphere and have a large size.

2.3.5.10 Brushing-and-Linking

Brushing-and-linking technique allows the synchronization among different views that present the identical dataset (Figure 2.31). When engineers make a change in any view, the change is dynamically reflected in the other views. This helps to represent the multidimensional data. The user may simultaneously analyze data with different points of view in order to find more meaningful information than independently analyzing the views.

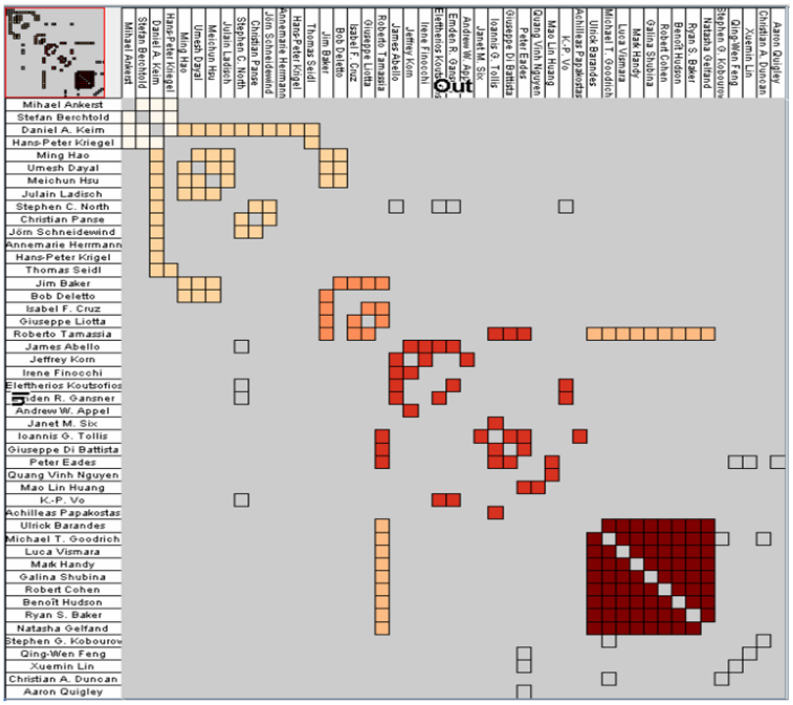


Figure 2.32: Matrix-based visualization for social networks (produced by [80]). Rows and columns represent persons and colored cells represent the social relations between the persons. Colors encode communities.

2.4 Related Visualizations

This section introduces the visualizations that are related to our work.

2.4.1 Matrix-based Visualization

The matrix-based visualization [23, 24] is commonly used for representing the array data whose relations are respectively represented in rows and columns. An important property of the matrix is the orderability [22–24, 174, 175]. The matrix layout is widely used in many domains: telecommunication [1], software architecture [17, 52, 196], social network [80–82, 197] (Figure 2.32), biology [114, 115].

2.4.1.1 Interactions on Matrix Visualization

Rao et al. [153, 158, 189] provided *Table Lens* technique by applying the Focus+Context concept with DOI distortion and semantic zooming to a table that represented large-scale tabular data (Figure 2.33). It may reduce the required display space by shrinking the row height of the table. Data values were graphically represented by bars inside cells that need less display space and is more intuitive than texts. In order to exactly identify the data values, users were allowed to flexibly enlarge the focused rows and read the detailed values in textual form. This way, table lens maintained a good overview of the data by graphical representations while providing detailed values for the focused data. Table lens technique was extended

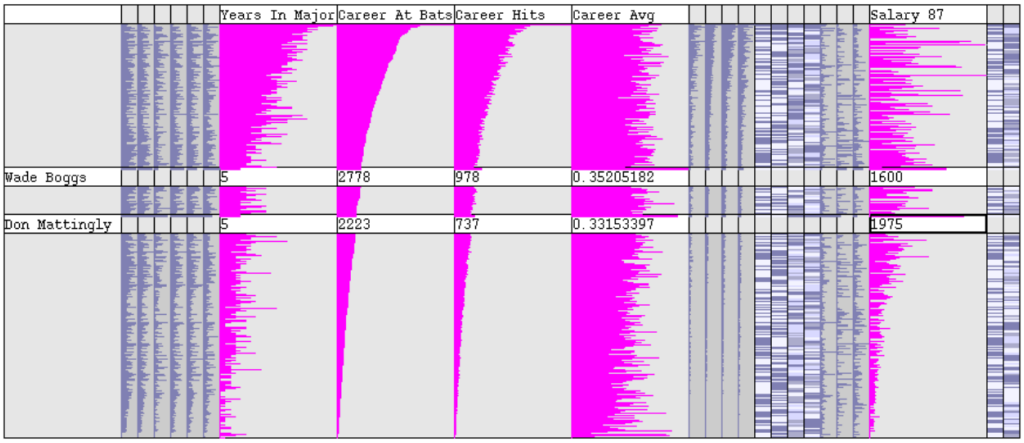


Figure 2.33: Table Lens (produced by [158]). Row height is shrunk in order to reduce the required display space. Bars graphically represent the data values. Detailed text data of the focused rows are presented in the enlarged rows.

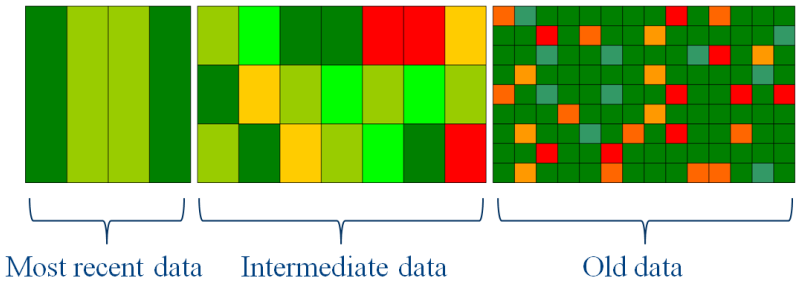


Figure 2.34: Multi-resolution of matrix (concept from [74]). Cells represent the time-series measurements of CPU utilization. Colors encode the results of the measurements.

by combining visualizations and interactions according to different application aspects [104, 115, 179, 188].

Scaling (or Zooming) of matrix layout shrinks or enlarges the row height as well as column width in order to balance overview and detail of the visible content. Flexible scaling may enhance the use of matrix layout. The scaling concept that uniformly changed the row height or/and column width was proposed in the work [1, 61, 196]. Hao et al. [72–74] proposed a matrix visualization for time-series data with the scaling based on the degree of interest (DOI) (Figure 2.34). Cells of the matrix represented the measurements of CPU utilization. The cells were arranged according to the time series. Colors encoded the results of the measurements. The DOI concept was used to determine the size of the cells with respect to their occurrence time. Frank van Ham [196] proposed a matrix with a multilevel scaling concept for representing the call-relations among components of large software systems. A cell represented a call between two components or self-calling of an identical component. The semantic zooming may be used for the matrix to present the detailed calling relations between sub components of two components. Graham et al. [71] proposed a matrix visualization for the project matching between users and potential partner for forming a project. The detail data of the assessors can be displayed using tables

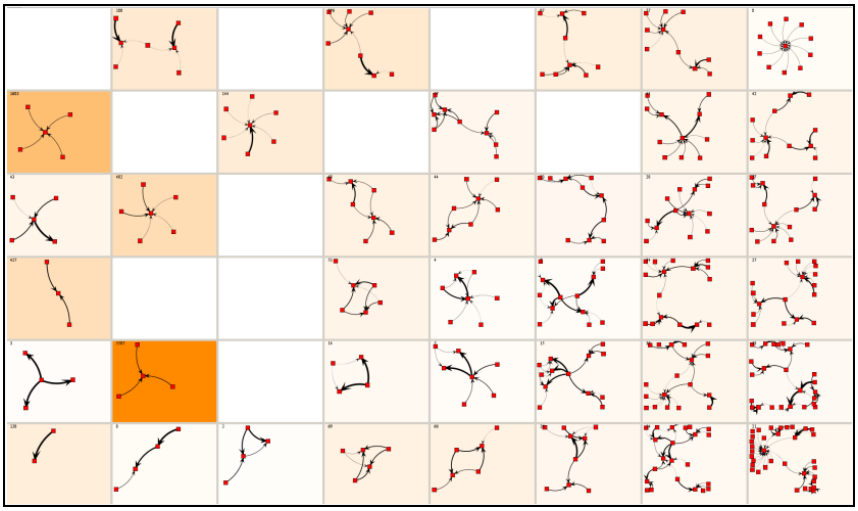


Figure 2.35: Nearest neighbor graphs are embedded in cells of a correlation matrix for representing results of clustering (produced by [202]).

inside cells. With the help of the semantic zooming concept, the visualization system shows a set of detailed relations between the specific data.

2.4.1.2 Hybrid Layout

The matrix-based visualization can be combined with other visualizations. Scatter plot matrix [36] embedded 2D scatter plots in each matrix cell in order to represent multivariate data. Each embedded plot represented two- or more-dimensional data depending on the corresponding cell. Von Landesberger et al. [202] embedded node-link diagrams that represented the nearest neighbor graphs into matrix cells in order to represent clustering results (Figure 2.35). Shen and Ma [167] used lines to link matrix cells for representing complex paths. Henry et al. [81] added curved lines on the table head of a matrix-based visualization in order to represent the relationships among persons in social network.

2.4.2 Node-link Diagram

Node-link diagram (Figure 2.25) is a well-known visualization layout consisting of a set of nodes and a set of links. Nodes are connected using links in order to represent relations among nodes. The structure of the node-link diagram depends on the particular semantic meaning or the structural rules [182]. This diagram is suitable to represent trees and networks.

The node-link diagram is widely applied in many domains. Plaisant et al. [154] proposed the SpaceTree that provided a dynamic scaling for branches in order to adapt to the limited display space. Becker et al. [18] used node position to represent the geographical data on the map and used lines to represent the data transfer of the telecommunication network. Heer et al. [78] visualized social networks using the node-link diagram that the portraits of contacts were attached to nodes for the intuitive identification (Figure 2.36). The node-link diagram was also applied to decision trees in [145, 194, 205, 211].

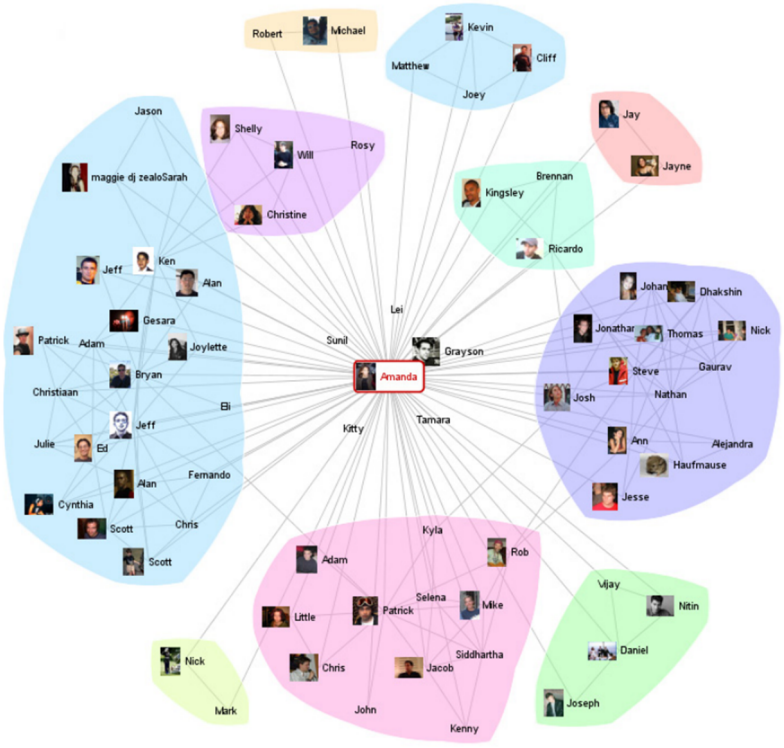


Figure 2.36: Node-link diagram (produced by [78]). This visualizes the social network where nodes represent persons and links represent social relations.

In many applications, the node-link diagram is combined with other visualization methods in order to represent complex data relations that are difficult to be visualized using a single layout. The matrix visualizations were integrated with the node-link diagrams in [165] and [82] in order to effectively reducing the line crossing. Zhao et al. [212] proposed a hybrid visualization where the treemap represented the hierarchies of large-scale data and node-link diagrams showed the topology of the data (Figure 2.37). Holten [86] proposed a composed visualization where the sunburst layout represented the hierarchies of the data and node-link diagrams represented relations among data objects. Boutin et al. [27, 28] applied silhouettes to node-link diagrams in order to represent the hierarchy information of networks. Hao et al. [75] presented a node-link diagram in a hyperbolic space in order to effectively investigate the interesting data objects in a large data warehouse. Jankun-Kelly and Ma [102] used a radial plot for representing the hierarchies of the network and used node-link diagrams for illustrating connections of nodes. The primary topology of networks were represented using node-link diagrams and the additional relations among nodes were represented by an additional set of edges in [84,123]. Shneiderman and Aris [171] represented the classified data objects in plots and used additional links to draw the relations between data objects.

2.4.3 Space-filling Representations

The commonly used layouts representing trees are node-link diagrams and space-filling diagrams (Figure 2.38). Space-filling diagrams use position or containment to

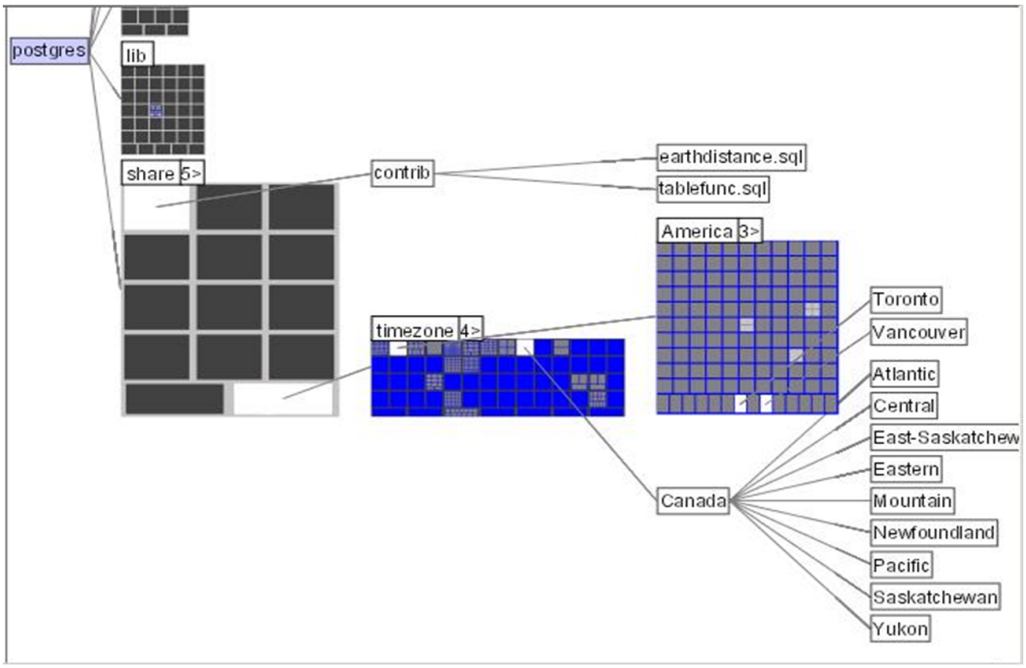


Figure 2.37: Elastic Hierarchies combines treemaps and node-link diagrams (produced by [212]).

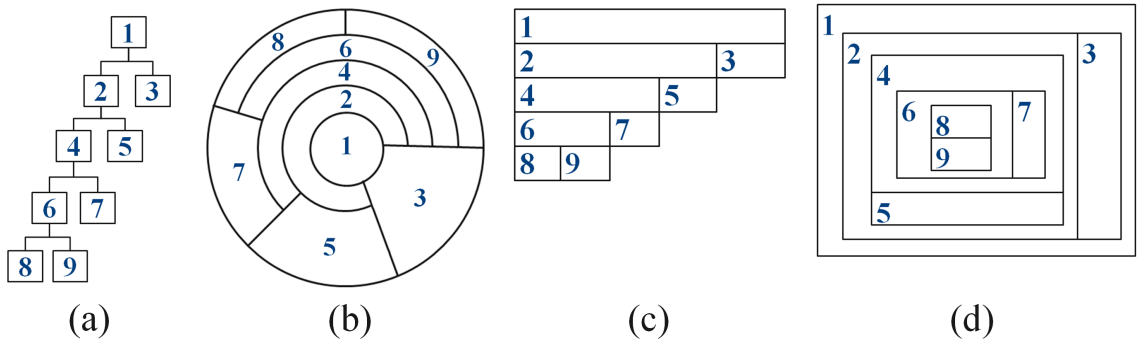


Figure 2.38: Tree visualization (concept from [13]). Different layouts represent identical data. (a) Node-link diagram (organizational chart). (b) Sunburst Layout (tree ring). (c) Icicle diagram. (d) Treemap.

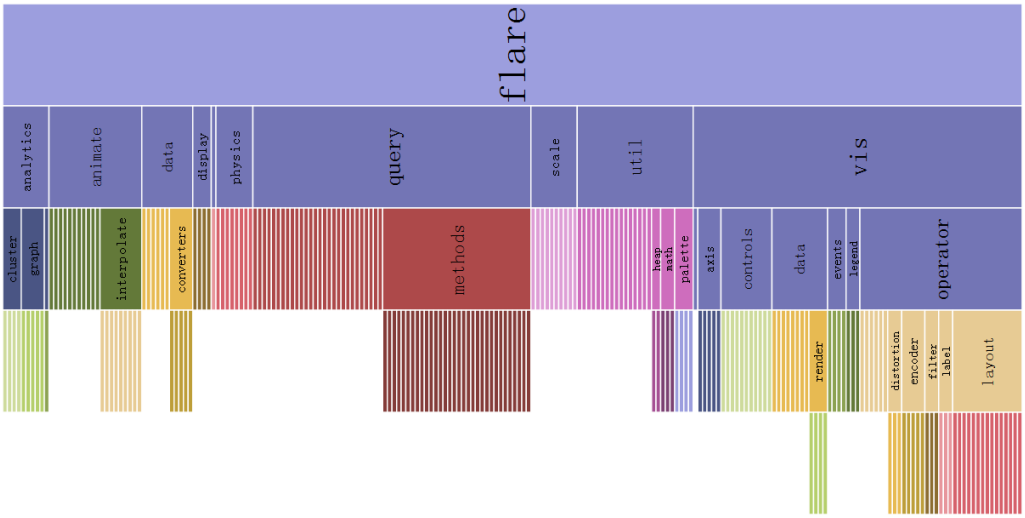


Figure 2.39: Icicle diagram representing the file system (produced using Protovis [155]).

represent the hierarchical relations instead of edges of node-link diagrams. Space-filling techniques provide better space-efficiency than node-link diagrams while they are not good at representing the topology of trees. This section we introduce the space-filling techniques: treemap, sunburst layout, and icicle diagram.

2.4.3.1 Icicle Diagram

The icicle diagram (also called icicle plot) provides a compact visualization layout for representing the tree structure [117, 121]. It uses stacked rectangles to represent the data hierarchy (Figure 2.38 (c), Figure 2.39). The child rectangles are placed under the parent rectangle. The length of leaf rectangles may be proportional according to a quantitative attribute of leaf nodes. According to the length, leaf nodes may be intuitively estimated and compared by this applied attribute. The length of a rectangle is the total of the length of its child rectangles. Thus, the rational relations between data sets may be represented by length of rectangles. When the length of a leaf rectangle is identical, the length of a non-leaf rectangle reflects the number of its child leaf nodes. Icicle diagram may have either vertical or horizontal orientation.

There are applications of the icicle diagram in various fields: Icicle diagrams were applied to decision trees [9, 14, 128] (section 2.4.4). In medical research, Wongsuphasawat et al. [207] represented the patient care data using icicle diagrams where hierarchies represented the sequence of the treatment and each hierarchy aggregated the patients at the same treatment step. Chevalier et al. [39] proposed a visual approach for C++ code analysis using the icicle-diagram-style syntax tree. Sifer [173] used icicle diagram to represent the hierarchical data websites, e.g., page structure, file type.

2.4.3.2 Sunburst Layout

The sunburst (also called "tree ring") is a radial space-filling technique for tree structure (Figure 2.38(b), Figure 2.40). Root is at the center and hierarchies are

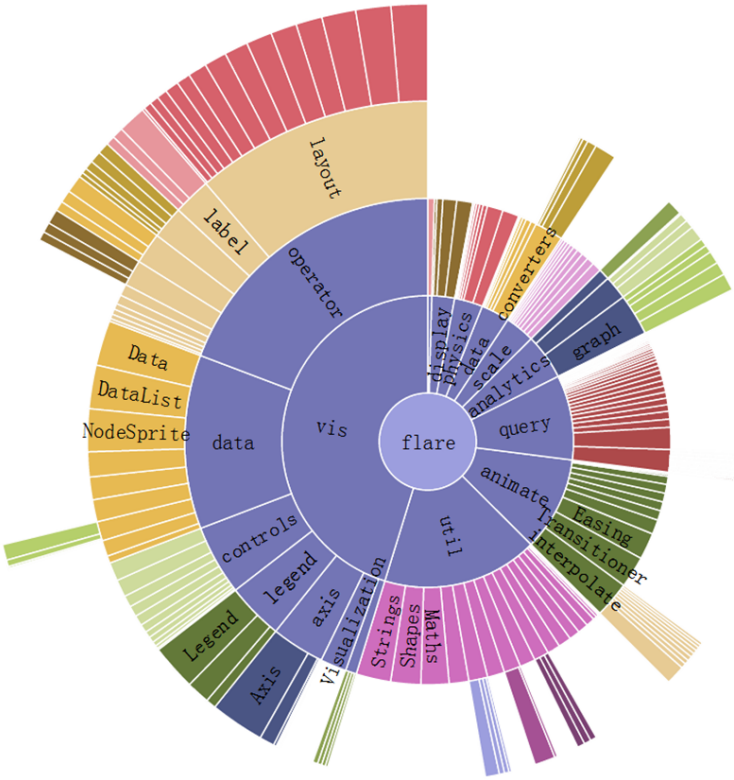


Figure 2.40: Sunburst layout representing the file system (produced using Protovis [155]).

represented as concentric circles. It can be treated as the radial icicle layout using polar coordinates instead of Cartesian coordinates.

Andrews et al. [7] used two linked semi-circle sunburst layouts for the file system, one represented the system overview and another represented the desired subsystem. The *InterRing* project [208] provided various interactions for the sunburst layout, particularly multi-focus distortions. Stasko and Zhang [180] represented hierarchies of file systems using the sunburst layout and used colors to encode file types. Keim et al. [112] provided a visualization approach for monitoring the data flow of computer networks that sunburst hierarchies respectively represented the source, destination, and other attributes, e.g., message. Patton et al. [151] visually analyzed the security of computer networks by representing the hierarchical clustered attack data using the sunburst layout. Herbert et al. [83] proposed a color-filled sunburst layout to visualize the data of the economic model "Minority Game".

2.4.3.3 Treemap Layout

Treemap layout represents the hierarchical data by recursively subdividing the given rectangle area [169] (Figure 2.38 (d), Figure 2.41). Treemap efficiently utilizes the space and provides a good overview of the hierarchical data. The quantitative attribute of the leaf nodes is represented using areas. There are various algorithms for treemaps, e.g., squared treemaps [30], ordered treemaps [172], modifiable treemaps [200], nested treemaps [105], quantum treemaps [21], cascaded Treemaps [131], and

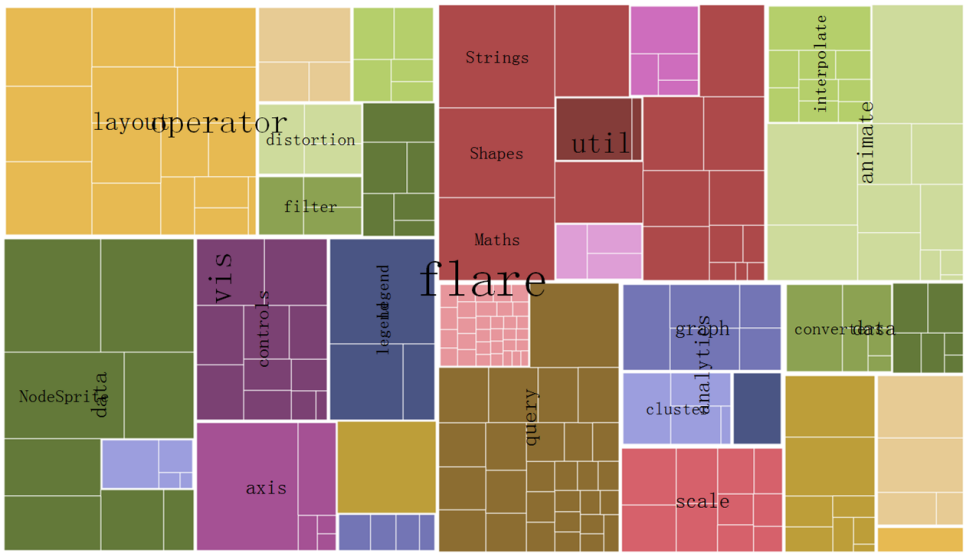


Figure 2.41: Treemap layout representing the file system (produced using Protovis [155]).

cushion treemaps [198]. Balzer et al. proposed the Voronoi Treemaps [12] that used polygonal shape, e.g., triangle or circle, instead of the traditional rectangular shape for more flexible adaptation of treemaps.

The treemap was applied in different domains. Heitzmann et al. [79] provided a treemap visualization for representing file systems and additionally used colors to encode user access permissions. Holten et al. [87] visualized the hierarchical elements of a software system using the treemap for analysis of system metrics. Lommerse et al. [130] proposed a visualization method for software analysis that represented the nesting of source code syntax using the treemap. Jin and Banks [103] visualized the competition trees on a tennis match using the treemap. The treemap were also used for representing the gene ontology data in biology research [11] and the stock-market data [106, 206].

2.4.4 Decision Tree Visualization

A decision tree is a tree-like predictive model that iteratively partitions a dataset to subsets according to partitioning rules. The root is the original data. The tree edges represent the rules and the non-root nodes represent the outcomes of different rules. By defining a set of partitioning rules, the desired subsets can be identified. A decision tree contains two kinds of basic information: the overall tree structure and the information attached to each node.

The decision tree applications focus on effectively and efficiently building and exploring a decision tree [190]. The point of building a decision tree is that users are able to quickly identify which node should be partitioned. Exploring a decision tree may facilitate an understanding of the decision-making process. The visually enhanced decision tree provides helpful information to support further decisions.

The decision tree is widely used for facilitating decision-making in various domains [8, 9, 14, 190]. Many systems provide visualized nodes of decision trees using

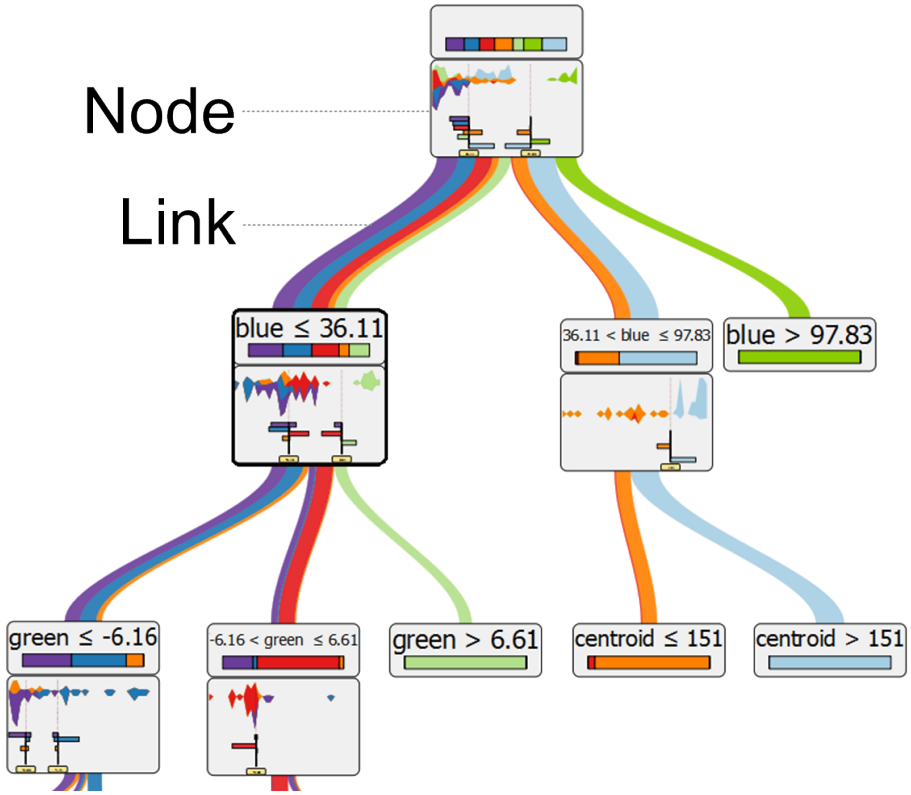


Figure 2.42: The node-link decision tree representing data classification (produced by [194]). Data are visualized and integrated in the nodes for supporting the split.

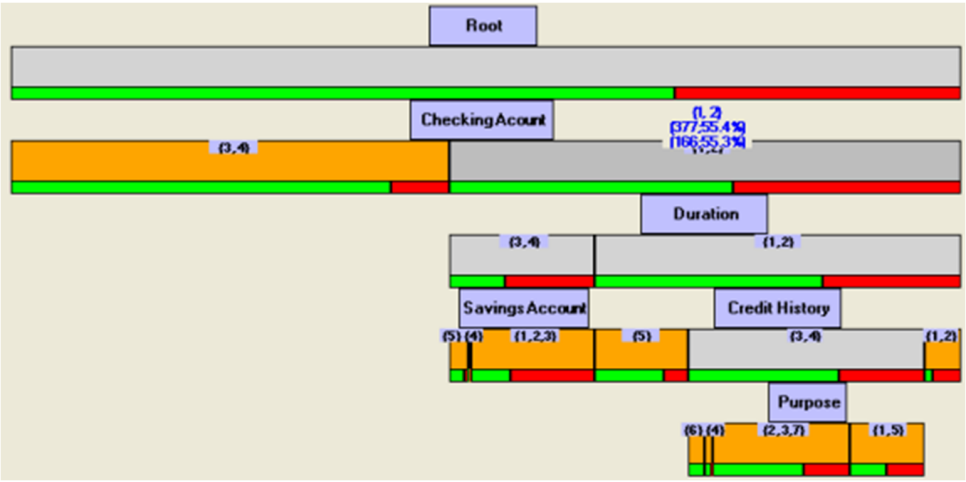


Figure 2.43: The icicle-style decision tree representing the classification process of credit data (produced by [128]).

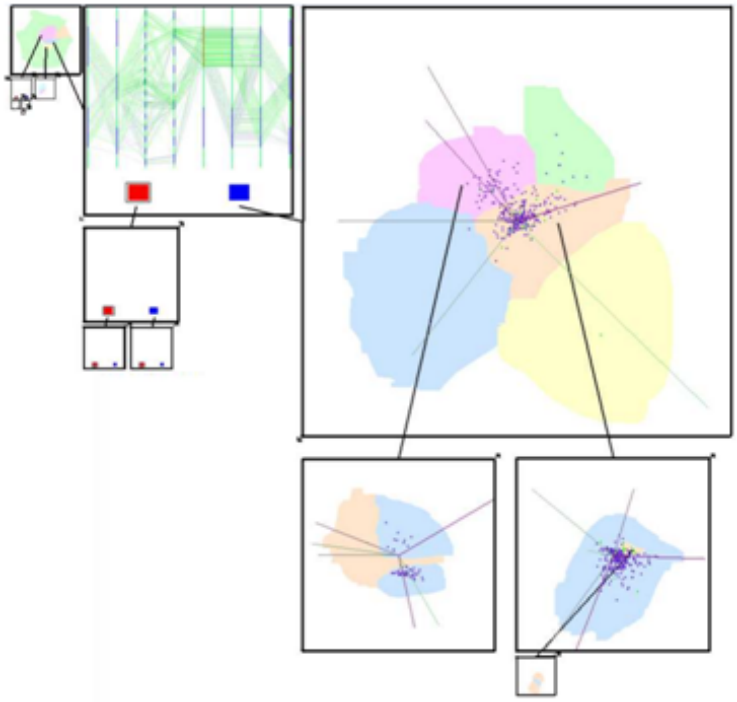


Figure 2.44: Decision tree associated with star coordinates and parallel coordinates for representing the classification results (produced by [190]).

properties, such as colors and bars. The simplest visualization way is the indentation diagram that represent the decision tree using a Windows Explorer-like layout. For example, Ankerst et al. [8, 9] provided a decision tree to present the partitioning process in data mining. The decision tree arranged the partitioning steps. Nodes represented the attributes that were used for splitting the data. Colors qualitatively encoded the classes of data.

The node-link diagram is widely applied to the decision tree visualization that provides an intuitive tree layout [145, 205, 211]. Van den Elzen et al. [194] used a top-to-bottom decision tree for data classification (Figure 2.42). The nodes represented the steps of the classification process. The resulting data of any step were visualized and attached to the corresponding node in order to support the next step of the classification. Pham et al. [152] used a sunburst layout to organize the decision tree for the visualization of machine-learning algorithm. The icicle diagram was applied to decision trees in [9, 14, 128]. This can provide a clear hierarchical structure without using much display space. The icicle diagram can effectively represent the node size according to the quantitative values of nodes. Liu and Salvendy [128] applied the icicle-style decision tree to the visualization of the classification process (Figure 2.43). Project PaintingClass [190] integrated parallel coordinates and star coordinates with a decision tree for exploring classified multi-dimensional data (Figure 2.44). The root node represented a visual projection of the data in the training set of the classification process. A non-terminal node worked on a projection that maps from multi-dimensional space into a two-dimensional display. Barlow et al. [14] proposed a visualization concept consisting of linked views to represent the decision information in data-mining process.

Chapter 3

Motivation

3.1 Problems and Objectives

This chapter describes the problems of the ordinary representation concepts for the analyses based on the FT and CFT. The objectives of our work are to address the problems by applying suitable visualization approaches.

3.1.1 MCS Analysis

MCSs are commonly represented using plain text and tabular forms (section 2.2.2). The representation methods show only the basic MCS information and a lot of hidden information is not visible. Engineers need to discover it by themselves. When the amount of MCSs is larger, e.g., one thousand MCSs, the ordinary representations reduce the efficiency of the MCS analysis. The problems of the ordinary representations of MCSs are summarized as follows and examples corresponding to these problems are demonstrated in section 3.2.1:

- **Weak representations for the relations between MCSs and basic events.** Engineers cannot effectively analyze the specific basic events with respect to the MCS analysis. In this case, engineers do not know what failure scenarios of the system may manifest when the basic events occur. The logical relations between the basic events included by a specific MCS cannot intuitively represented. Example 1 in section 3.2.1.1 depicts the problem.
- **Unsatisfactory overview of a large-scale MCS dataset.** Engineers can view only a small part of the MCSs and cannot explore patterns based on the overview of MCSs. In addition, navigation of MCSs is not efficient with the lacking context. Example 1 in section 3.2.1.1 describes the issue.
- **Inefficient identification of the important information.** Without suitable indications, engineers need to manually identify the important information from MCSs. The identification may take a great deal of effort, particularly when the number of MCSs is large. Example 2 in section 3.2.1.2 shows the problem.
- **Weak analysis for the failure propagation.** The views of MCSs and the CFT structure are separate. In this case, engineers need to frequently switch views in order to identify the failure propagation of MCSs because of the

loss of the suitable association. Additionally, because of the nesting of the CFT components (described in section 3.1.2), engineers need to sequentially go through the parent CFT components for identifying a specific basic event. Example 3 (section 3.2.1.3) illustrates the problem.

Our objective is to design a visualization approach that facilitates the analysis of MCSs. The effective analysis based on relations between MCSs and basic events should be possible. Pattern analysis based on the overview of a large number of MCSs should be possible. Engineers should be allowed to quickly identify the important information whose detail should be possibly investigated. In addition, engineers should be able to effectively analyze the failure propagation of MCSs while maintaining the overview of MCSs as the context information.

3.1.2 Importance Analysis

For the importance analysis, the importance values of basic events are usually represented in data-aggregated forms, e.g., tables and charts (section 2.2.3). The issues of the representation concepts are briefly summarized as follows and application scenarios demonstrate the issues in section 3.2.2.

- **Weak association between the importance of basic events and the FT (or CFT) structures.** The data-aggregated forms provide a summary view of the importance of basic events while the FT (or CFT) structure represents the logic relations between basic events. There are few integrations between the data-aggregated forms and the FT (or CFT) structure. In two cases, engineers have to frequently switch views in between:
  - analyze the failure propagation along the FT (or CFT) structure with respect to the important basic events that are identified in the data-aggregated forms.
  - assess the importance of the specific basic events when focusing on the FT (or CFT) structure.

  Example 1 in section 3.2.2.1 depicts the problem.
- **Weak patterns analysis.** Overview of the importance of basic events may represent patterns. Engineers cannot effectively associate the overview with the FT (or CFT) structure for identifying further patterns. An important pattern is the distribution of the important basic events over the FT modules (or CFT components). This allows the critical FT modules (or CFT components) that contain the important basic events to be analyzed. An additional pattern is the logic relations between the important basic events. Example 2 in section 3.2.2.2 demonstrates the issue.
- **Few effective concepts for representing the deeper-nested CFT components.** We apply the CFT in our work because of its advanced properties, particularly the corresponding relation between the CFT component and the system architectural component. Thus, we discuss the representation issues of the CFT rather than those of the traditional FT. Due to the nesting of CFT

components, when locating a specific basic event, engineers need to sequentially go through all the direct and indirect parent CFT components of the basic event until the basic event is found. It may waste much effort if the basic event is included by a deeper-nested CFT component. This issue hampers the analysis for the failure propagation of the important basic events, which depicts the effects of the occurrence of the important basic events. Additionally, because the structures of CFT components are displayed in separate views, it is difficult for engineers to obtain a continuous critical path between the desired basic event and the top event. It is particularly ineffective when simultaneously analyzing multiple critical paths. Example 3 in section 3.2.2.3 illustrates the problem.

As a goal, we need to design a visualization approach that associates the importance of basic events with the CFT structure. Engineers should be able to analyze patterns, particularly with respect to the critical CFT components. Our approach needs to facilitate the identification of the vulnerable system architectural components corresponding to the critical CFT components. In addition, the analysis for influence of the important basic events should be possible, which includes the identification of the influenced CFT components and the analysis of the failure flow along the CFT structure.

3.1.3 The Safety Improvement Process

The ordinary representation concepts of the safety improvement process separate the relevant data across several individual views (section 2.2.4). The significant data are mostly presented in data-aggregated forms, e.g., tables. There are also particular views, such as FT structures, and the decision trees. The views may be linked and interactively show the data of the improvement process. The issues of the representation concepts are summarized as follows and depicted using examples in section 3.2.3.

- **Weak integration between the sequence of modifications and the detailed data of modifications.** It is difficult for engineers to investigate the detailed data of the specific modification, e.g., cost of the modification, when focusing on the overall sequence of modifications in the decision tree. In this case, engineers have to switch to other views for the required data. In the whole analysis process, the view switching is possibly very frequent and requires a great deal of additional effort.

- **Few visualizations of the significant data associated with modifications.** The textually represented data are not intuitive for understanding modifications, particularly for comparing modifications.

- **Few interactive associations between data attached to the improvement solutions and the FT (or CFT) structure.** Engineers cannot interactively analyze the logic function of the basic event corresponding to the specific modification. Additionally, the effect of a design modification cannot be intuitively reflected. In this case, engineers cannot quickly analyze the

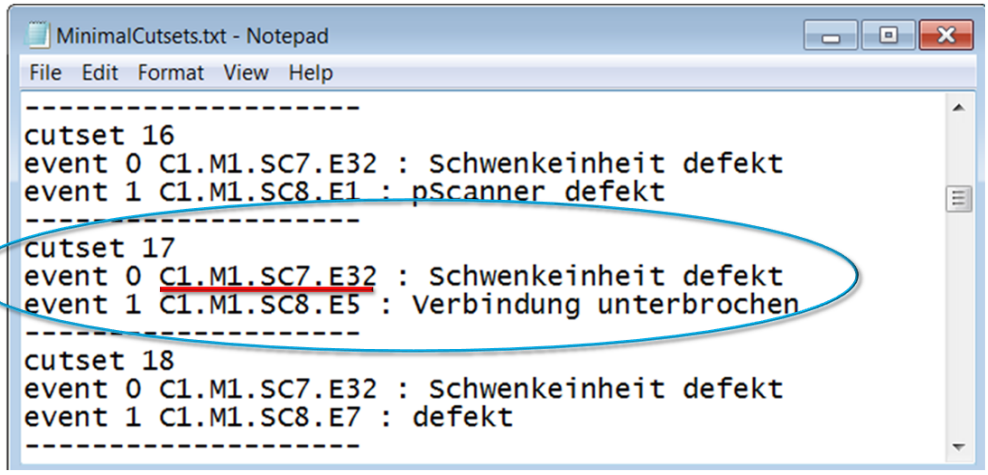


Figure 3.1: Example 1 for the MCS analysis (produced using [193]). The identification of the partner basic events for the specific basic event "C1.M1.SC8.E5". The first identified partner basic event "C1.M1.SC7.E32" is included by MCS "cutset 17".

impacts upon the intermediate failures along the critical path of the corresponding basic event. When simultaneously analyzing multiple modifications, the issue will be more critical.

Our goal is to design a visualization approach that can graphically represent the significant data and integrate the graphical representations into the safety improvement process. Engineers should be able to quickly investigate the detailed data of modifications while maintaining the sequential relations between the modifications. In this way, engineers may focus on the analysis process without needing to waste energy for the data look-up. Additionally, patterns of the safety improvements with respect to the CFT components should be explored. For example, the distribution of the basic events involved by a specific solution over the structure of the FT (or CFT).

3.2 Examples

The issues of the currently used representation concepts for the MCS analysis, the importance analysis, and the safety improvement process are summarized in section 3.1. In this section, application scenarios are demonstrated to describe the issues.

3.2.1 Examples of MCS Analysis

3.2.1.1 Application Scenario 1 for the MCS Analysis

In many cases, identifying the related MCSs and related basic events for a specific basic event is a significant aspect in the MCS analysis [136]. The result of the analysis shows in which scenarios this basic event may cause the top event to occur. For example, for a given basic event "E1" that is critical and difficult to improve.

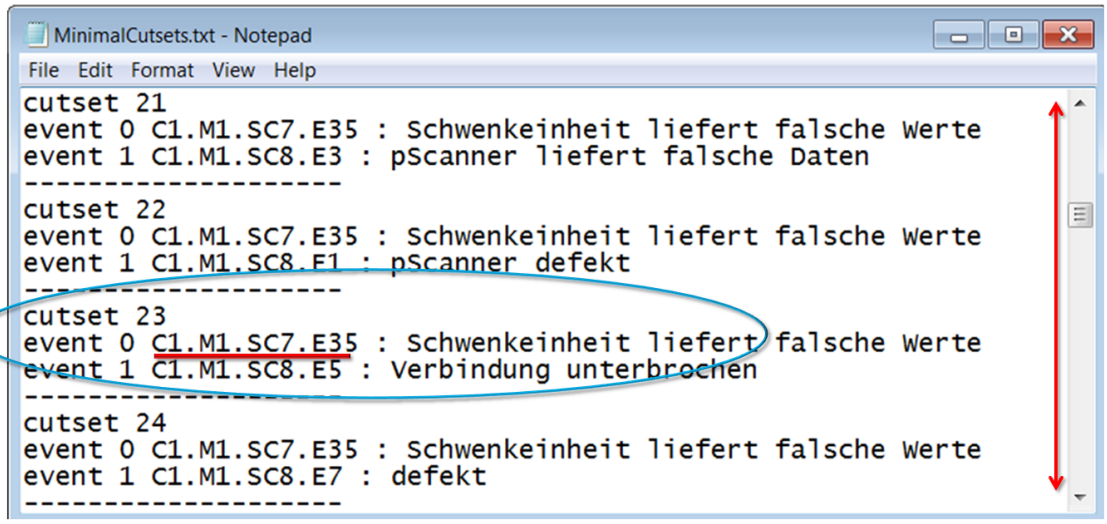


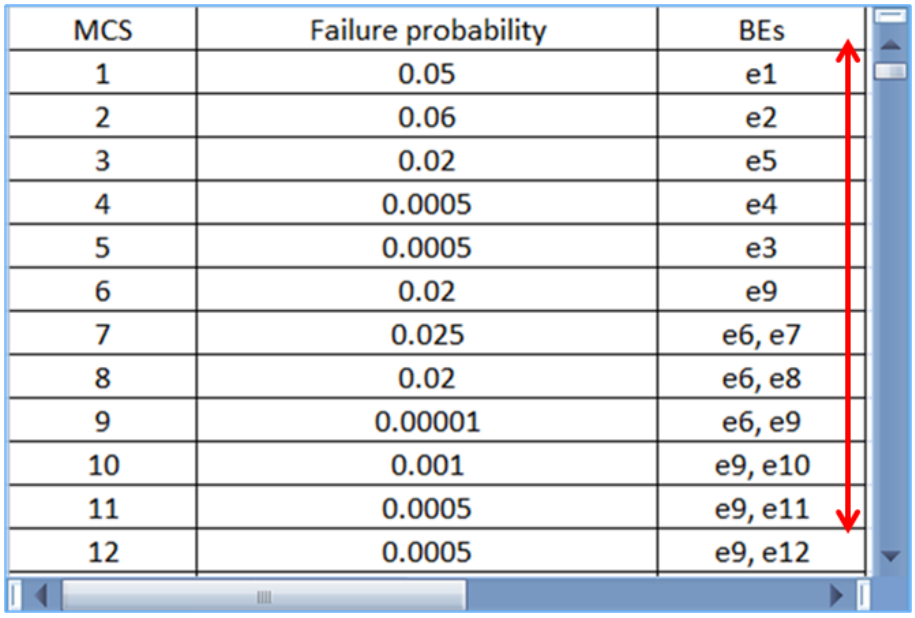
Figure 3.2: Example 1 for the MCS analysis (produced using [193]). The second resulting MCS is identified after scrolling the text. It includes the partner basic event "C1.M1.SC7.E35".

The related MCSs are: MCS1 = {E1, E2} and MCS2 = {E1, E3}. In this case, engineers need to concentrate on addressing the partner basic events, i.e., "E2" and "E3". Usually engineers can not get support from the ordinary representations for this analysis.

An application scenario is presented with respect to this issue. Additionally, the scenario also demonstrates the issues of the ordinary representation concept when analyzing a large number of MCSs. In this scenario, all 118 MCSs are represented using plain text (Figure 3.1 and Figure 3.2). The basic event "C1.M1.SC8.E5" is critical and difficult to address. The objective of the example is to identify the MCSs that include the given basic event and the partner basic events. We achieve the objective by the following steps:

1. We identify the first MCS including the given basic event. The MCS "17" is identified as a result (Figure 3.1).
2. We analyze this MCS and find that it includes the partner basic event "C1.M1.SC7.E32".
3. We scroll the text in order to identify more relevant MCSs. The MCS with the ID "23" is then identified. It includes the basic event "C1.M1.SC7.E35" (Figure 3.2).
4. We repeat the identification from step 1 to 3 until all the relevant MCSs and their basic events are found. Finally, we find all 39 MCSs and 66 basic events.

According to the application example, we conclude that the textual forms do not intuitively represent the relations between the specific basic events and MCSs. Tabular forms have the same problem because table columns do not represent the fixed basic events, but the index of elements (i.e., basic events) of each MCS, i.e., a column may present different basic events for different MCSs. Another shortcoming is that the textual forms do not provide a satisfactory overview of a large number



MCS	Failure probability	BEs
1	0.05	e1
2	0.06	e2
3	0.02	e5
4	0.0005	e4
5	0.0005	e3
6	0.02	e9
7	0.025	e6, e7
8	0.02	e6, e8
9	0.00001	e6, e9
10	0.001	e9, e10
11	0.0005	e9, e11
12	0.0005	e9, e12

Figure 3.3: Example 2 for the MCS analysis. There are 880 MCSs overall in the data table.

of MCSs. We have to frequently scroll the text for navigating and identifying the desired MCSs. In this case, engineers need additional effort for the analysis process.

3.2.1.2 Application Scenario 2 for the MCS Analysis

In order to preferentially address the major failure scenarios, it needs to assess the criticality of MCSs. The commonly used representation concepts provide very few highlighting methods for the estimation of MCSs. An example is presented in order to depict the problem. In this example, MCSs and the associated data are represented in a table (Figure 3.3). We define the criteria for the criticality of the MCSs. MCSs are treated as the most critical MCSs when they have failure probabilities larger than "0.01". MCSs are treated as the secondary critical MCS when they have failure probabilities between "0.01" and "0.001". The goal is to identify and classify the most critical MCSs and secondary critical MCSs.

In order to fulfill the goal, we manually compare the textual failure probability of each MCS with the thresholds for estimating the criticality of the MCS. As a result, we finally find 55 most critical MCSs and 120 secondary critical MCSs. Without intuitive representations for the criticality of MCSs, it takes additional effort for estimating MCSs by investigating the textual values. Based on the example, we conclude that it is inefficient to identify the desired MCSs using the ordinary representation methods because of the loss of suitable highlighting methods.

3.2.1.3 Application Scenario 3 for the MCS Analysis

This example describes the shortcomings of the ordinary representation concepts with respect to the analysis of the failure propagation. MCSs and the CFT components are represented over separate views and there are few context of MCSs.

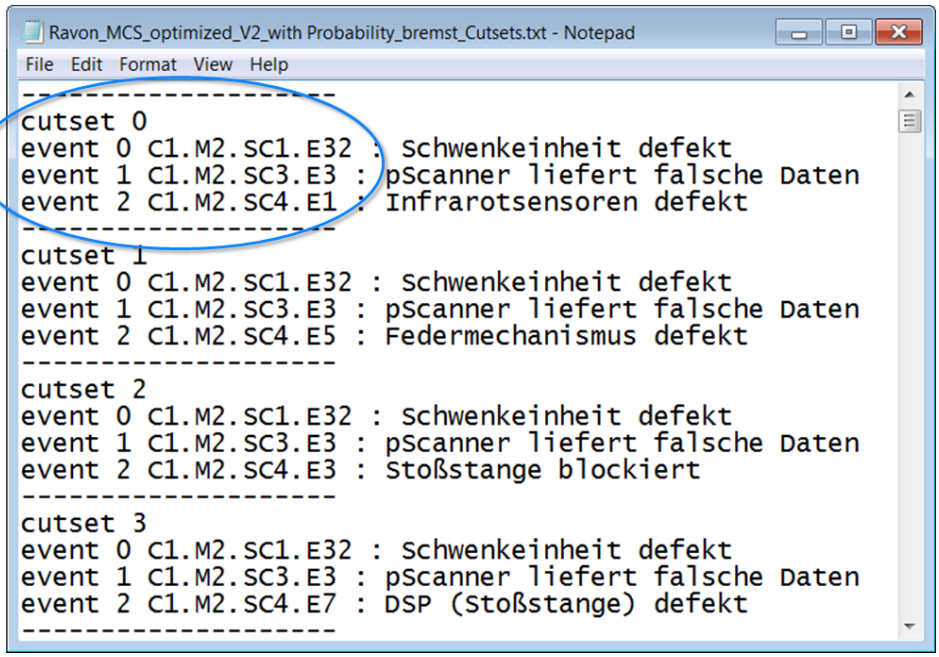


Figure 3.4: Example 3 for the MCS analysis. Data table of MCSs.

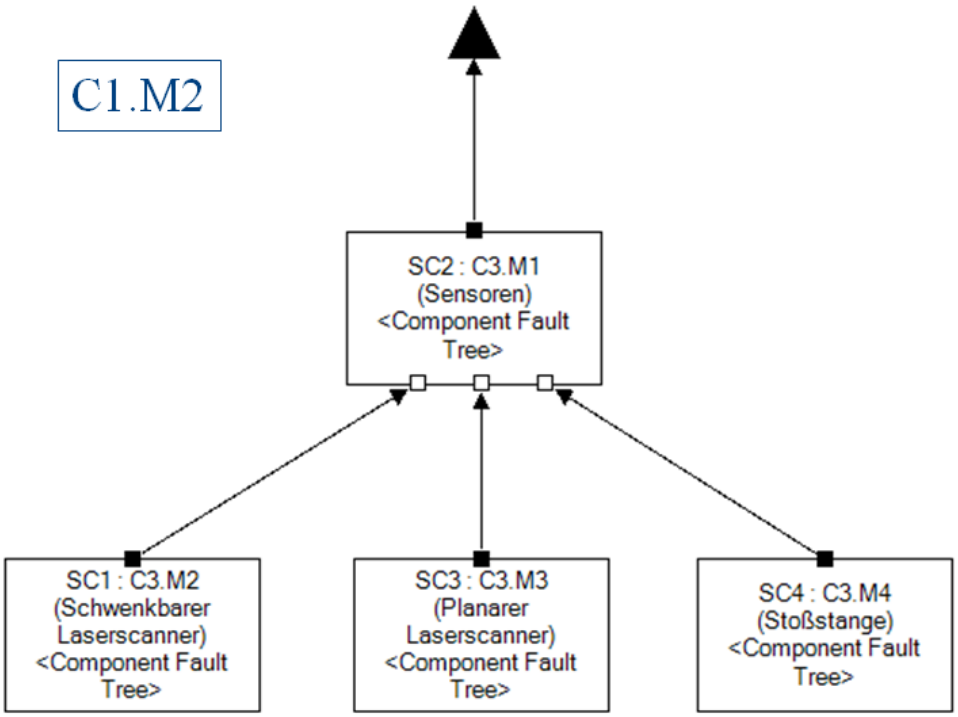


Figure 3.5: Example 3 for the MCS analysis (produced using [193]). The system level CFT component “C1.M2”.

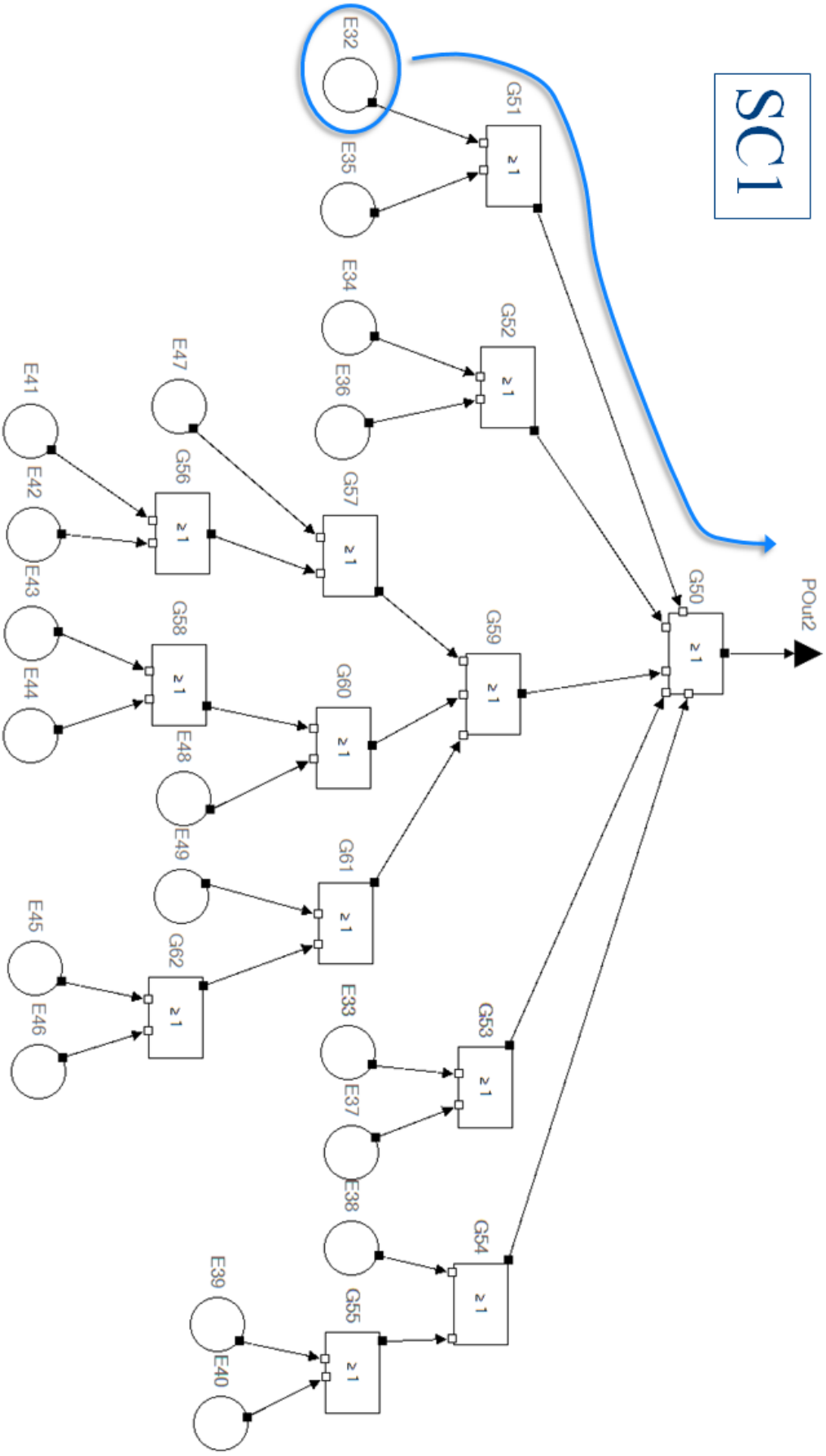


Figure 3.6: Example 3 for the MCS analysis (produced using [193]). The sub-CFT component "SC1".

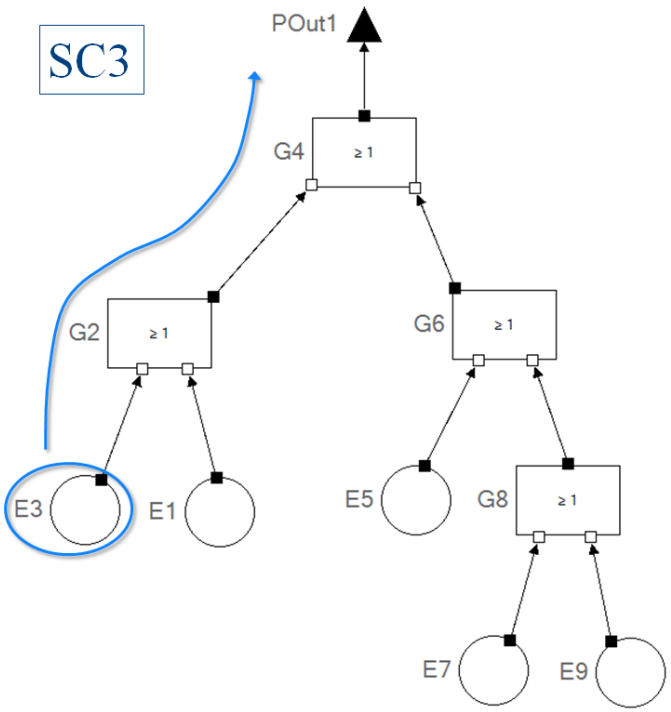


Figure 3.7: Example 3 for the MCS analysis (produced using [193]). The sub-CFT component "SC3".

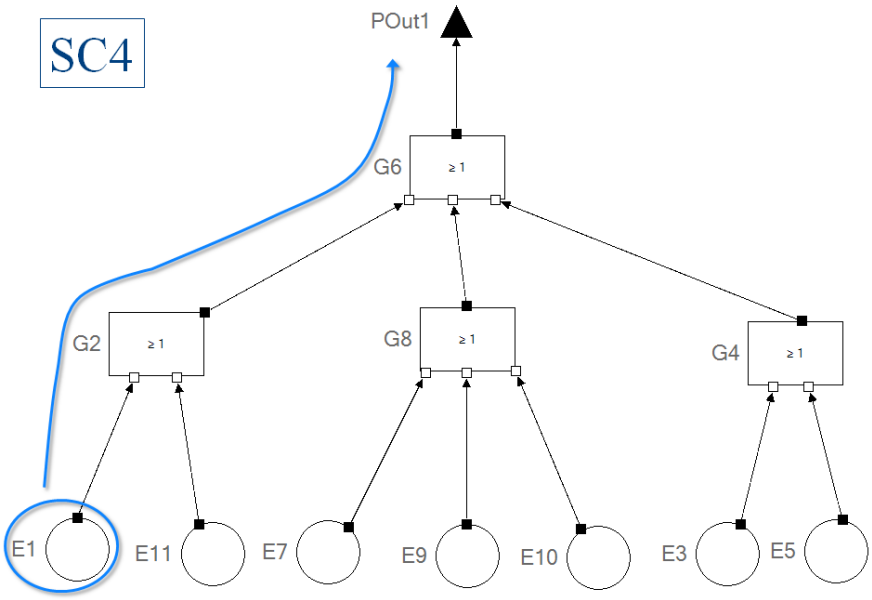


Figure 3.8: Example 3 for the MCS analysis (produced using [193]). The sub-CFT component "SC4".

In addition, due to the nesting relations of the CFT components, for identifying a specific basic event, engineers have to sequentially go through all the direct and indirect parent CFT components.

The objective of the example is to identify the critical paths of basic events that are included by the MCS "cutset 0". We only investigate the failure propagation inside the direct parent CFT components of the basic events in the example. We present the example using the tool ESSaRel [193]. We perform the following steps to accomplish the objective:

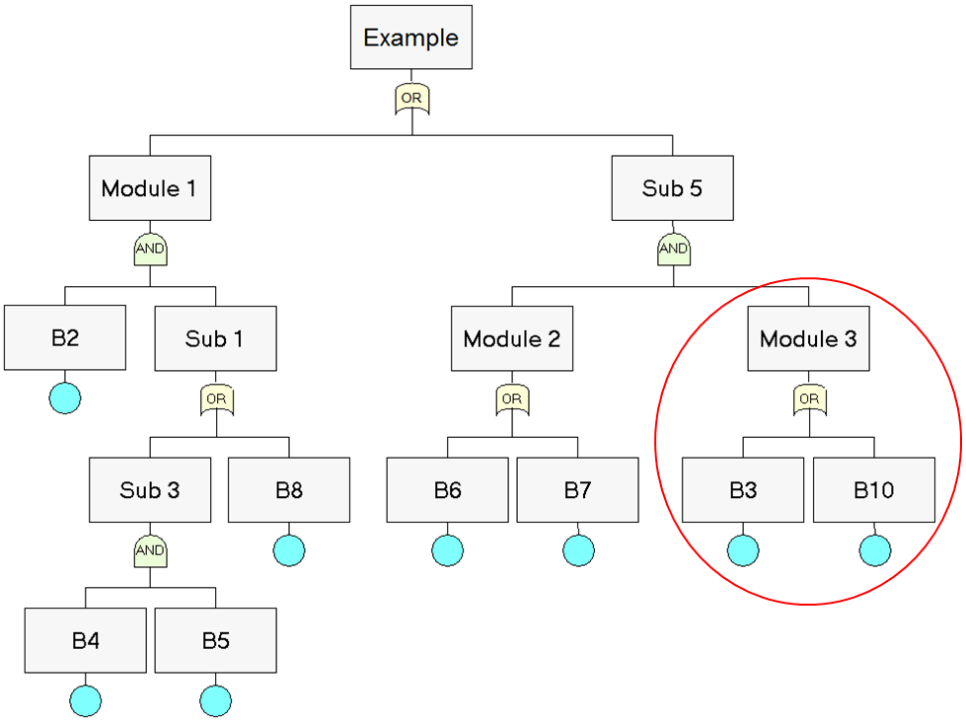
- We analyze the MCS "cutset 0" using the text editor. We find that this MCS includes three basic events (Figure 3.4). The first one has the ID "C1.M2.SC1.E32". According to the naming rules, the ID of a basic event comprises of the IDs of the parent CFT components and its ID (i.e., the last part). Thus, we know that the first identified basic event is directly contained by component "SC1" that is the sub-component of component "C1.M2".
- We switch views to the system-level component "C1.M2" (Figure 3.5).
- By double-clicking the rectangle symbol representing the sub-component "SC1", we switch to an individual view displaying the logical structure of the CFT component (Figure 3.6).
- We locate the basic event "E32" and its critical path.
- We switch to the text editor (Figure 3.4). We find that the second relevant basic event is "C1.M2.SC3.E3".
- We switch to the system-level component "C1.M2" again (Figure 3.5).
- By double-clicking the rectangle symbol representing component "SC3", we switch to an individual view displaying the logical structure of the CFT component (Figure 3.7).
- We locate the basic event "E3" and its critical path.
- We switch to the text editor (Figure 3.4). We find that the last relevant basic event is "C1.M2.SC4.E1".
- We switch to the system-level component "C1.M2" again (Figure 3.5).
- By double-clicking the rectangle symbol representing component "SC4", we switch to an individual view displaying the logical structure of the CFT component (Figure 3.8).
- We locate the basic event "E1" and its critical path.

By using the ordinary representation concept, we have to switch different views between MCSs and CFT components, and between different CFT components. Additionally, when afterwards investigating the logical relations between the three basic events, we have to additionally investigate component "SC2" (the upper sub component in Figure 3.5). The frequent view switching takes a great deal of efforts.

3.2.2 Examples of the Importance Analysis

3.2.2.1 Application Scenario 1 for the Importance Analysis

In many cases, engineers purpose to assess the importance of desired basic events when they focus on the logic structure of a FT. The ordinary representation concepts



(a) FT structure. "Module 3" contains two basic events: "B3" and "B10".

N	Code	Occur-rence	Q mean	FV Imp.
1	B2	2	0.02	0.928298
2	B8	1	0.05	0.920208
3	B10	2	0.0005	0.0598126
4	B7	2	0.08	0.0441697
5	B6	2	0.05	0.0276061
6	B3	2	0.0001	0.0119627
7	B4	1	0.02	0.008098
8	B5	1	0.022	0.008098

(b) Table for the importance of basic events.

Figure 3.9: Example 1 for the importance analysis (produced using [6]). We need to switch views between the FT structure and the data table in order to assess the importance of the basic events that are identified in the FT structure.

of the FT structure do not directly represent the importance of basic events. This example describes this issue. The goal of the example is to compare the importance of basic events of the module "Module 3". We fulfill this goal by the following steps:

1. We identify the first basic event in "Module 3" of FT (Figure 3.9). The resulting basic event is "B3".
2. We switch to the table for the importance of basic events and investigate the importance of the basic event "B3". The result is 0.0119627.
3. We switch views to the FT structure again for identifying the next basic event of "Module 3". The resulting basic event "B10" is identified.
4. We switch back to the table and find that the importance of "B10" is 0.0598126.
5. We compare both basic events and conclude that the basic event "B10" is more important than the basic event "B3".

In the process of the example, it takes much more effort to frequently switch views. The issue will be more critical when a lot of basic events in the FT structure need to be compared. This problem also occurs by CFTs in the same way.

3.2.2.2 Application Scenario 2 for the Importance Analysis

Importance analysis based on FTs (or CFTs) usually provides the summary information and the possible ranking of basic events according to the importance of basic events. When analyzing the importance of basic events by associating the summary information and FT structures, engineers may identify patterns with respect to the distribution of importance of basic events over the FTs (or CFTs). For example, which modules (or CFT components) contain the important basic events, and which important basic events are contained in a specific module.

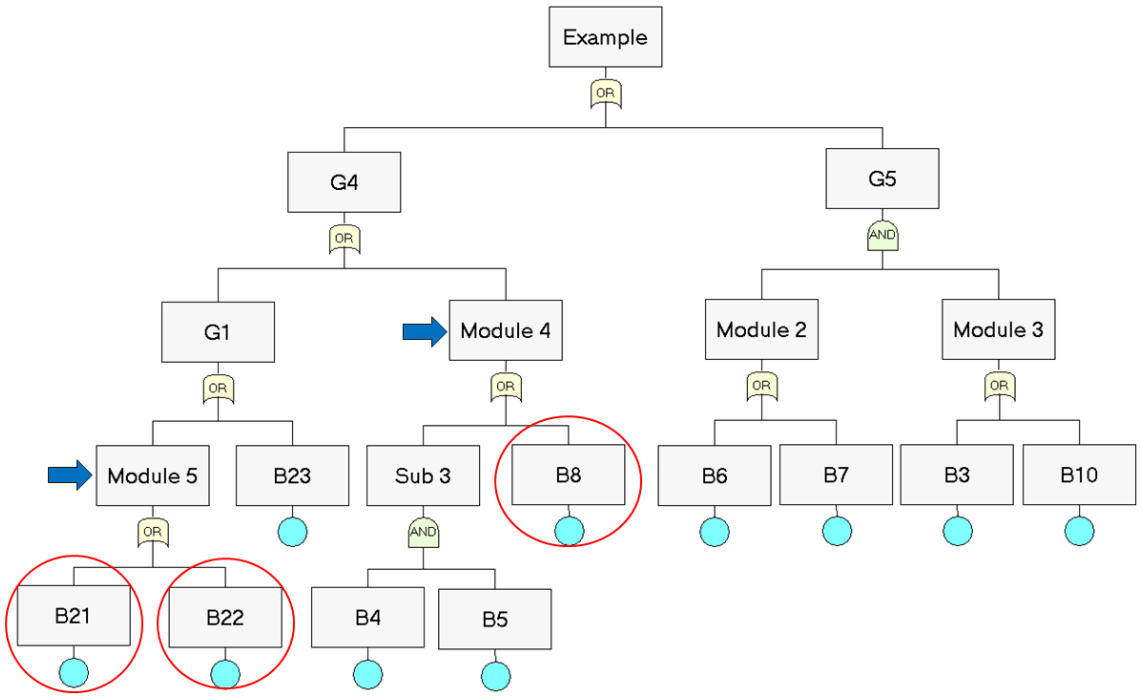
We provide an example of depicting the issue. In the example, the important basic event is defined as a basic event that has an importance value greater than 0.04. The importance of basic events are summarized in a table and ranked in descending order. The goal is to identify the distribution of the important basic events over the modules of a FT and the logical relations between the important basic events. The following steps are needed in order to achieve the goal:

1. We view the table for the importance of basic events (Figure 3.10 (a)) and find that the basic event "B22" is the most important.
2. We switch to the FT structure and locate the basic event "B22" (Figure 3.10 (b)). We find that "Module 5" contains the basic event. We manually mark the basic event up in the FT structure.
3. We repeat step 1 and step 2 to identify other important basic events and locate them in the FT structure.

As a result, according to the marks we made in the process, we can obtain a distribution of the important basic events in the FT structure. we find that "Module 5" contains the basic events "B21" and "B22", and "Module 4" contains the basic event "B8". Basic events "B21" and "B22" are connected by an OR-gate.

N	Code	Occur-rence	Q mean	FV Imp.
1	B22	1	0.25	0.573405
2	B21	1	0.2	0.458724
3	B8	1	0.05	0.114681
4	B23	1	0.01	0.0229362
5	B4	1	0.02	0.001009
6	B5	1	0.022	0.001009
7	B10	2	0.0005	0.000149
8	B7	2	0.08	0.000110
9	B6	2	0.05	6.88e-005
10	B3	2	0.0001	2.98e-005

(a) Data table for the importance of basic events.



(b) FT structure. The basic event "B8", "B21", and "B22" are the important basic events.

Figure 3.10: Example 2 for importance analysis (produced using [6]).

According to the example, we conclude that by the commonly used data-aggregated forms, it is difficult to effectively reflect the distribution of basic events over the FT structures. In the process, we need to manually mark up the resulting basic events that we found in the FT structure, otherwise we will forget them when afterwards analyzing other important basic events. In this case, it is difficult to intuitively analyze the logical relations between multiple important basic events in the FT and the distributions of them over the FT unless these basic events are manually marked. The issue becomes more serious for some advanced demands, such as an overview of critical paths of multiple important basic events. This problem is also serious for CFTs. This is even more critical because of the nesting relations and separate views of CFT components (described in Example 3.2.2.3) .

3.2.2.3 Application Scenario 3 for the Importance Analysis

In the CFT, when locating a basic event, engineers have to go through all the direct and indirect parent CFT components because of the nesting relations between CFT components. Additionally, it is difficult to obtain a continuous critical path because CFT components are separately represented over several views. The issues are illustrated using this application example. The goal of the example is to identify the most important basic event "C1.M1.SC18.SC4.E32" and analyze its critical path.

In the example, the ID of a basic event represents the location of the basic event. The ID of the important basic event "C1.M1.SC18.SC4.E32" may be divided into the following parts: "C.M1" is the system-level CFT component, "SC18" is the sub-component, "SC4" is the more deeper sub-component of "SC8", and "E32" is the local ID of the basic event in component "SC8". We need the following steps to achieve the goals:

1. We view the system-level CFT component "C1.M1" and find the rectangle symbol representing the (sub-)component "SC18"(Figure 3.11 (1)).
2. We switch to component "SC18" (Figure 3.11 (2)). We find the rectangle symbols representing the (sub-)component "SC4".
3. We switch to component "SC4" and find the desired basic event "E32" (Figure 3.11 (3)).
4. Then, we investigate the critical path of the basic event by tracing back from the view (3) to the view (1). We may separately review the partial critical paths in different views (Figure 3.12).
    - We identify the partial path along the structure of component "SC4" in the view (3): E32 → G5 → out-port. We need to analyze the parent components of "SC4" in order to identify the remaining path.
    - We switch to component "SC18" (the view (2)) that is the direct parent component of "SC4". We continue the path from the rectangular symbols representing component "SC4": G3 → out-port.
    - We finally go to the view of the system-level component "C1.M1" (the view (1)) that directly contains component "SC18": G6 → G1 → Top Event.

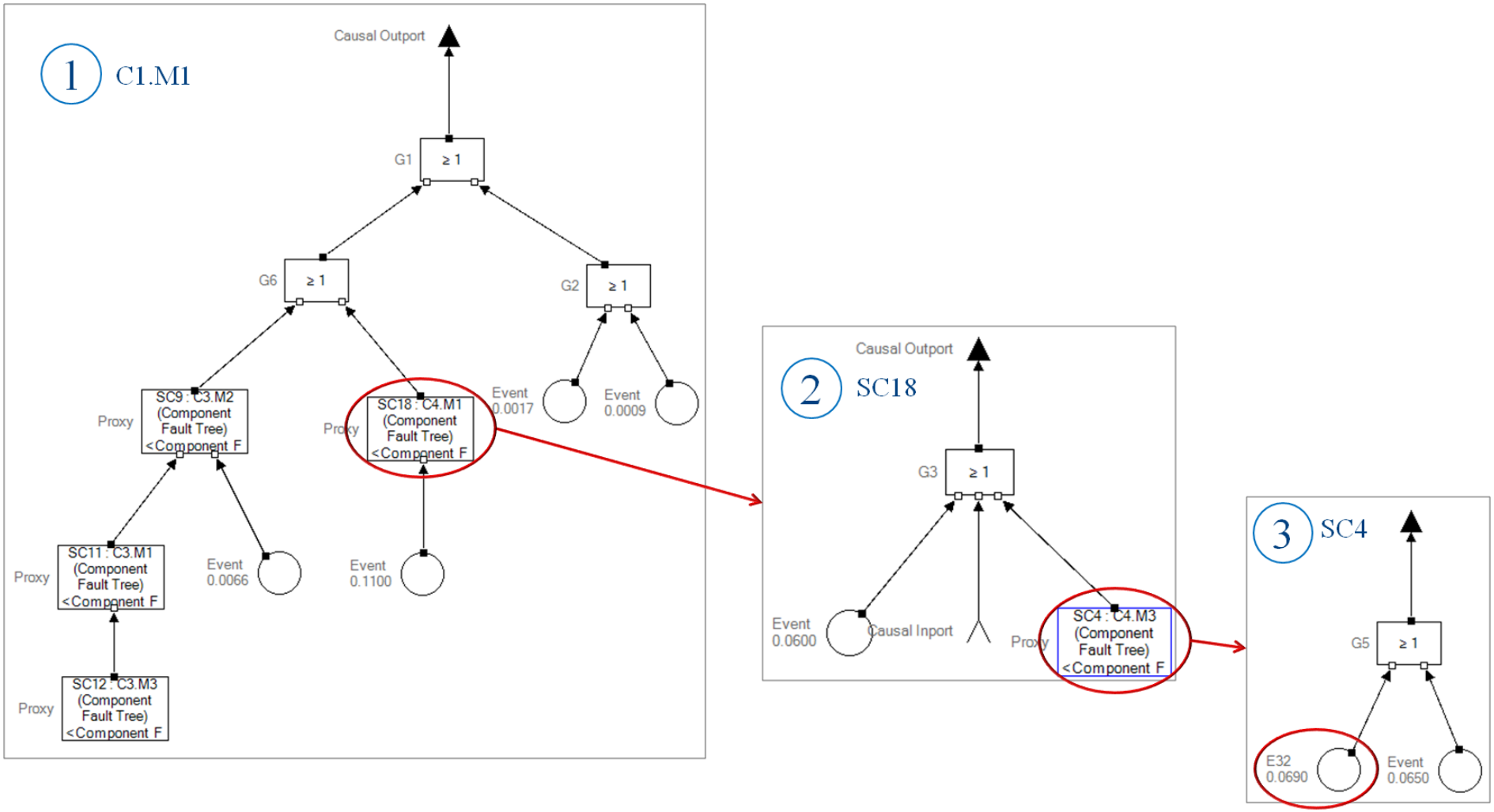


Figure 3.11: Example 3 for the importance analysis (produced using [193]. (1) The system-level CFT: "C1.M1". (2) The sub-CFT component: "SC18". (3) The sub-sub-CFT component: "SC4".

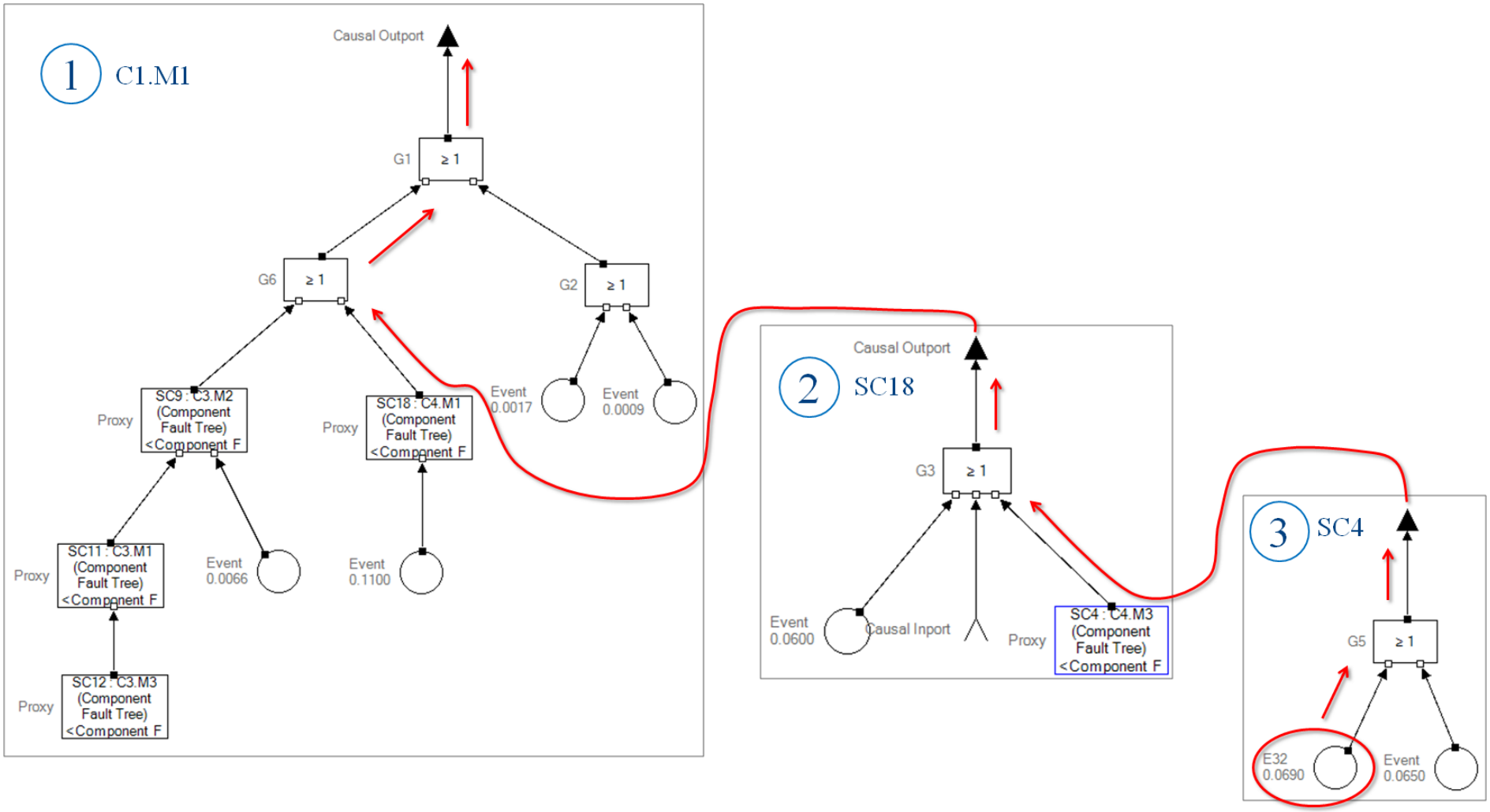


Figure 3.12: Example 3 for the importance analysis - 2 (produced using [193]. The critical path starts at the basic event “E32” in view (3) and ends at the top event in view (1) by way of component “SC18” in view (2).

The example depicts two main issues. First, in order to identify the basic event "E32", we have to sequentially go through its indirect parent components "C1.M1" and "SC18". By the ordinary representation of the CFT, it is difficult to directly view the basic event in its direct parent component "SC4" without considering the higher-level parent components. However, in some cases, when analyzing an important basic event, engineers only focus on its direct parent component because this basic event directly influences only this CFT component. In this case, investigating all the higher-level parent components wastes a lot effort. When analyzing an important basic event whose direct parent component is deeper nested in the CFT structure, the issue will be more critical. Second, we cannot analyze the continuous critical path in one view. The representation concept using separate views reduces the effectiveness of the analysis for the failure propagation of the basic events. When simultaneously analyzing multiple critical paths, the issue will be more serious.

3.2.3 Examples of the Safety Improvement Process

Because the ordinary representation concepts separate the data over individual views, engineers need to frequently switch views according to the associations between the data. In this way, efficiency and effectiveness of the safety improvement process is reduced because the significant context information is missing. The information related to the safety improvement process mainly involves two aspects: the construction of solutions and the analysis of solutions. The section presents the application scenarios with respect to these two aspects.

3.2.3.1 Application Scenario 1 for the Safety Improvement Process

The example illustrates the issues of the currently applied representation methods when constructing an improvement solution. The objective of the example is to identify a new optimal design modification in one iteration of the safety improvement process. The actual failure probability of the top event is "0.0976". We define the goal value of the improvement as "0.04". We need to identify the most cost-effective modification when there are multiple modification alternatives. We fulfill the objective by the following steps:

1. We perform the important analysis and analyze the resulting histogram of the importance of basic events (Figure 3.13). We find that the basic event "B22" is the most important.
2. There are two modification alternatives (according to the domain knowledge). We simulate the first design modification alternative "M1" that substitutes the old part having a failure probability of "0.06" with a new one having a failure probability of "0.02". The cost of this modification is 5 units. We update the FT according to the modification in order to simulate the change of the system risk (Figure 3.14). The failure probability of the top event is reduced from "0.0976" to "0.0592". The cost-effectiveness of the modification alternative is stored in a data table.
3. We simulate the second modification alternative "M2" that applies a parallel redundancy by adding one identical hardware component. Before performing

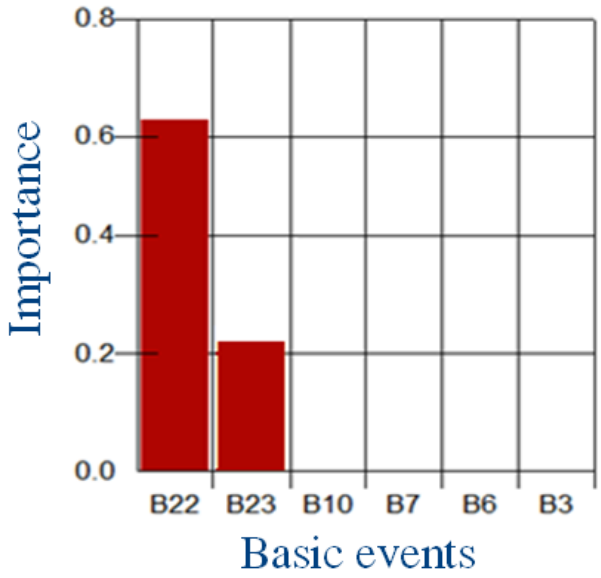


Figure 3.13: Example 1 for the safety improvement process (produced using [6]). Histogram represents the importance of basic events. The basic event "B22" is the most important basic event.

the simulation, we have to reset the FT to the unchanged initial state.

4. We update the FT structure according to the redundant concept (Figure 3.15). The updated failure probability of the top event is reduced from the initial value "0.0976" to "0.0435". The cost of the modification is 10 units.
5. We switch to the data table of the modifications "M1" (Figure 3.16 (a)). The cost-effectiveness is "130.21".
6. We switch to the data table of the modifications "M2" (Figure 3.16 (b)). The cost-effectiveness is "184.84".
7. We compare both values and finally find that modification "M2" is more cost-effective. Then we select the redundancy modification and refuse the substitution method.
8. We compare the new failure probability of the system and the goal value. The new failure probability "0.0435" is still larger than the goal value of "0.04". Thus, we need to perform further iteration of the system improvement starting from step 1.

According to the example, we conclude that the separated data views increase the efforts when constructing solutions. And the analysis may be interrupted when looking up data in different views.

3.2.3.2 Application Scenario 2 for the Safety Improvement Process

The example presents an application scenario focusing on the analysis of the constructed solutions. In this example, there are four existing solutions. The objective can be separated into three ordered sub-tasks:

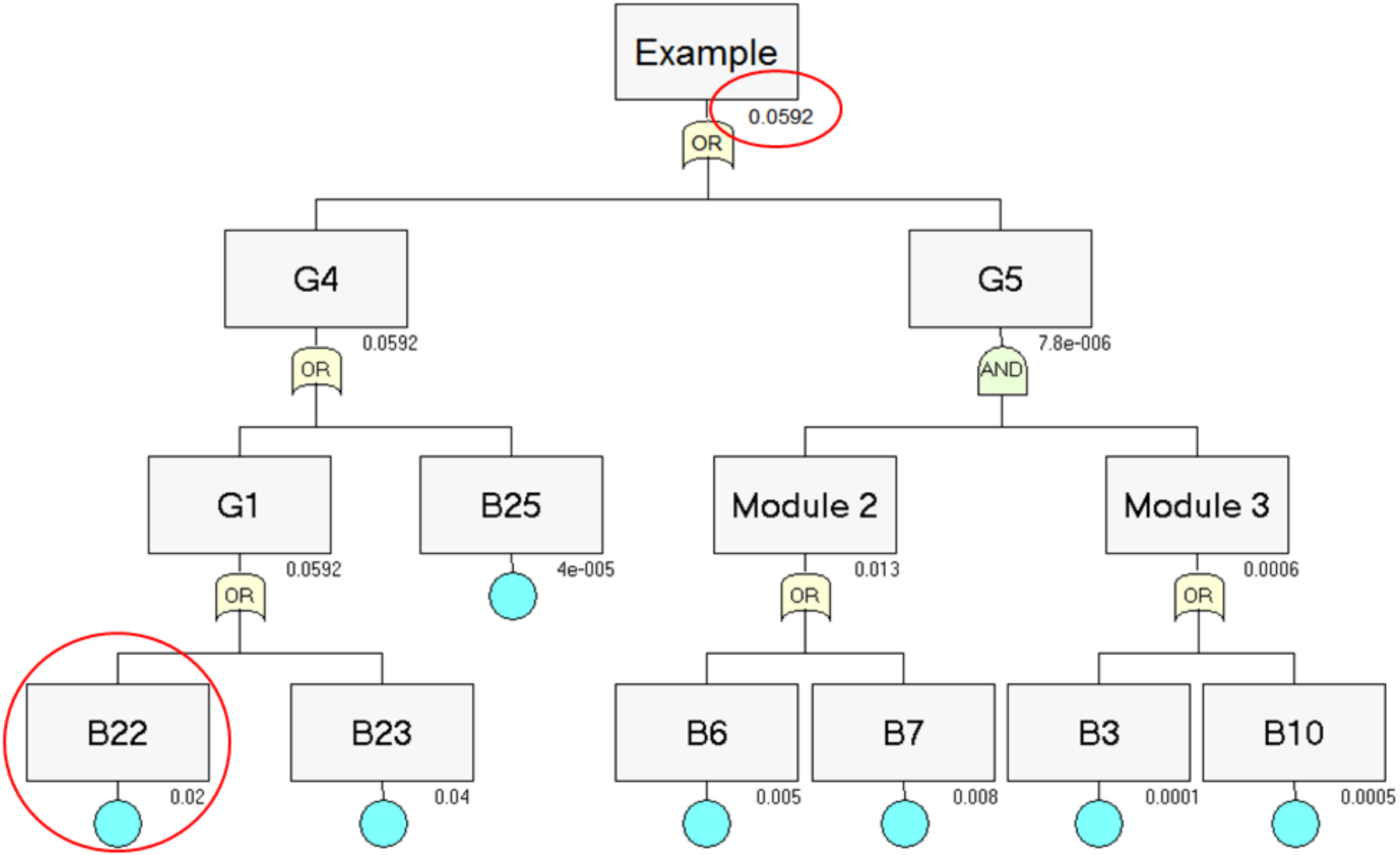


Figure 3.14: Example 1: the safety improvement process (produced using [6]). FT is updated by applying the substitution concept to "B22". The failure probability of the top event is reduced from "0.0976" to "0.0592".

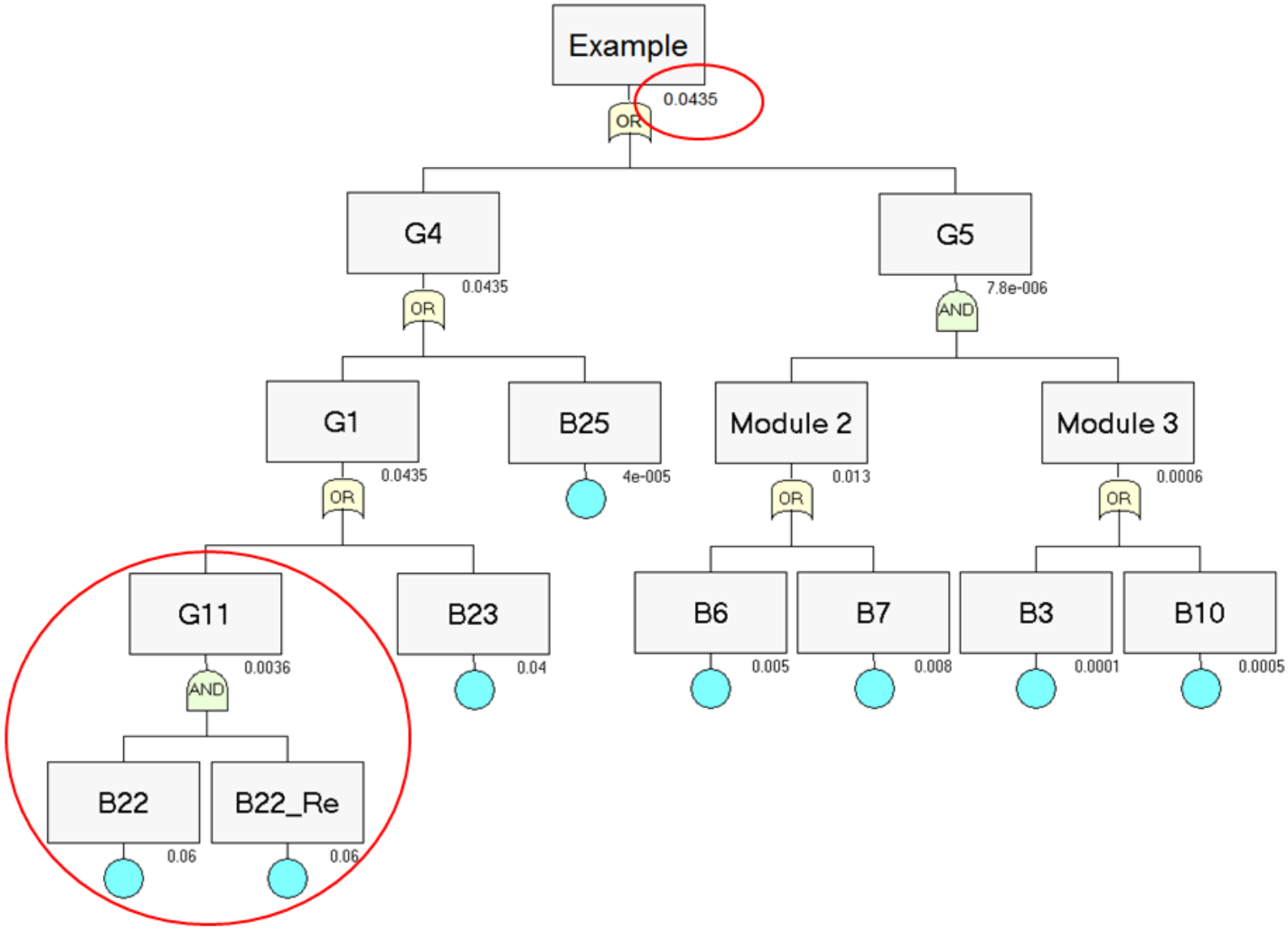


Figure 3.15: Example 1 for the safety improvement process (produced using [6]). FT is updated by applying the redundancy concept. A parallel redundancy is applied to "B22" by adding one identical hardware component to the system. The failure probability of the top event is reduced from "0.0976" to "0.0435".

Modification ID	M1
Type	substitution
Cost	5
Cost-effectiveness	130.21
Failure probability reduction of top event	0.0384
Basic Event	B22

(a)

Modification ID	M2
Type	redundancy
Cost	10
Cost-effectiveness	184.84
Failure probability reduction of top event	0.0541
Basic Event	B22

(b)

Figure 3.16: Example 1 for the safety improvement process. Data tables for the modifications "M1" and "M2". The cost-effectiveness of modification "M2" is larger. Hence, the redundancy concept is preferred.

ID	System Failure Probability	Total Cost
S1	1.00E-02	150
S2	1.00E-03	85
S3	1.00E-03	60
S4	1.00E-03	50

Figure 3.17: Example 2 for the safety improvement process. Data table for the improvement solutions. Solution "S4" has the minimal total cost.

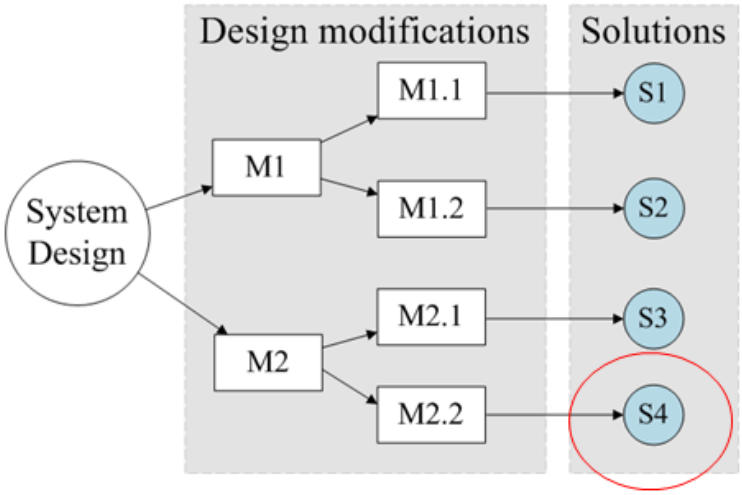


Figure 3.18: Example 2 for the safety improvement process. The decision tree represents the sequential modifications. Solution "S4" consists of the modifications "M2" and "M2.2".

Modification ID	M2
Type	substitution
Cost	5
Cost-effectiveness	130.21
Failure probability reduction of top event	0.0384
Basic Event	B10

(a)

Modification ID	M2.2
Type	substitution
Cost	2
Cost-effectiveness	166.67
Failure probability reduction of top event	0.012
Basic Event	B23

(b)

Figure 3.19: Example 2 for the safety improvement process. Data tables for different design modifications.

- Identify the solution having the lowest cost;
- Identify the modification, which causes the largest reduction of failure probability of the top event in the identified solution;
- Locate the related basic event in the FT structure.

We accomplish the objective by the following steps:

1. We view the data table summarizing solutions in order to identify the solution having the minimal total cost (Figure 3.17). Solution "S4" is identified as a result.
2. We switch to the decision tree view that arranges the modifications (Figure 3.18). As a result, we find that the modifications "M2" and "M2.2" belong to solution "S4".
3. We switch to the data table for modification "M2" (Figure 3.19 (a)). The failure probability of the top event is "0.0384".
4. We switch to the data table for modification "M2.2" ((Figure 3.19) (b)). The failure probability of the top event is "0.012".
5. We compare the probability values and find that "M2" causes the larger impact on the top event.
6. We switch to the data table for modification "M2". We identify that the corresponding basic event is "B10".
7. We switch to the FT structure view in order to locate the basic event "B10" and its critical path (Figure 3.20).

We conclude that reviewing the existing solutions using the separate views is an inefficient way. Frequently switching views may bother the analysis process. The textual values are not intuitive for the comparisons. We need to manually identify the basic events corresponding to the solution in the FT view. When we purpose to simultaneously investigate multiple modifications, even multiple solutions, the required efforts may exponentially increase.

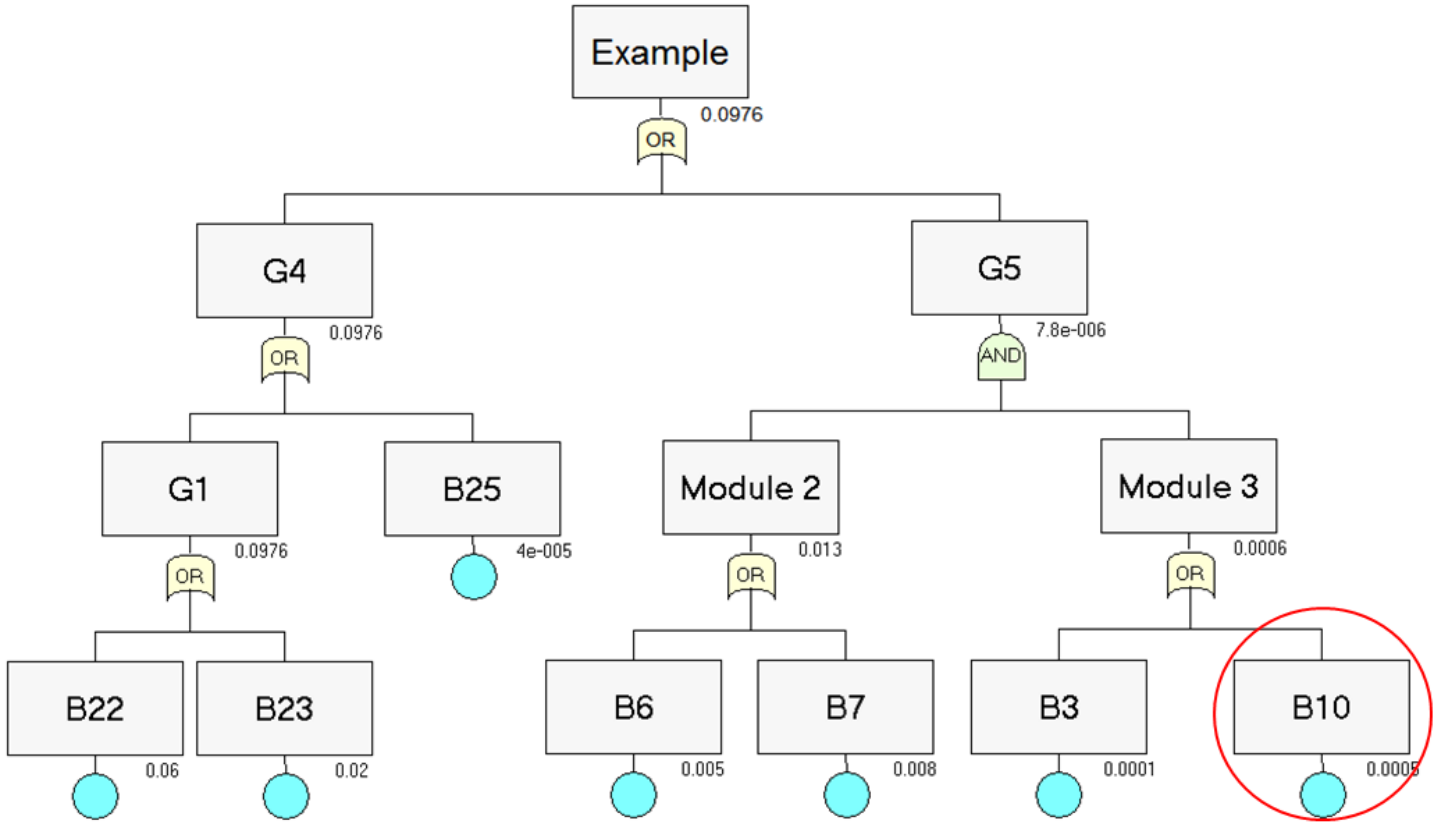


Figure 3.20: Example 2 for the safety improvement process (produced using [6]). We locate the basic event "B10" in the FT structure.

Chapter 4

Design Concepts for Visual Structures

In the dissertation, we propose visualization approaches in order to support three analyses based on the CFT analysis: the MCS analysis, the importance analysis, and the safety improvement process. Visual mapping is the core step of the reference model of information visualization (section 2.3.4), which transforms data to the fundamental structure of a visualization (i.e., a spatial layout). This chapter focuses on the visual mapping and conceptually discusses the design of the visual structures of our visualizations, particularly the spatial substrates. The details of graphical properties and interactions are discussed in Chapter 5, Chapter 6, and Chapter 7. The common design rules for our visualization approaches are defined with respect to the following aspects:

- Readability and space efficiency. The estimation of layouts has two significant aspects: readability and space efficiency. The readability describes how correctly and fast the user can find the answers from a visualization [69, 120, 146, 187, 213]. The space efficiency is aimed at the rational utilization of screen space. In our work, readability is the primary consideration of the design decision and the space efficiency is the meaningful complement.

- Dimensionality of visualization. The decision of the dimensionality is an open question in information visualization. Two-dimensional layouts are most commonly used. The third dimension is applied to the particular data that cannot be effectively represented using only two dimensions. The evaluations between 2D and 3D visualizations were discussed in [44, 46, 116, 125, 150, 166, 177, 181, 204] with respect to various aspects. However, the results were dependent on the tasks and settings of the evaluations, and thus were not identical. It is difficult to conclude the general principles for determining dimensionality. We basically prefer the 2D layout in our work, unless there are particular requirements for the third dimension.

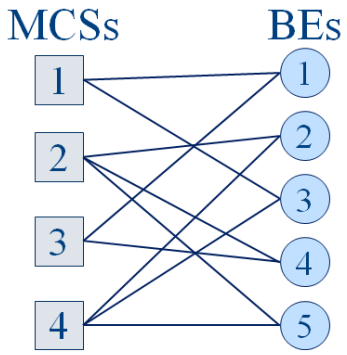


Figure 4.1: The relations between MCSs and basic events.

Element	N	MCS_1	MCS_2	...	MCS_n	BE_1	BE_2	...	BE_m
Type of Element	N								
Relation	Link								
Failure Probability	Q								
Unreliability level	O								
Number of relations	Q								

Table 4.1: The case-by-variables table for the MCS data.

4.1 Regarding the MCS Analysis

4.1.1 Data

According to the definition of the MCS analysis (section 2.1.3) and the motivations of our work (section 3.1.1), the required raw data (shorter form "R") of the analysis is collected with respect to the following aspects:

- Analysis of the inclusion relations between MCSs and basic events.
  - R1: the inclusion relations between MCSs and basic events.
- Estimation of MCSs and basic events.
  - R2: the failure probability of MCS.
  - R3: the failure probability of basic event.
- Investigation of the failure propagation of MCSs.
  - R4: the CFT structure.

4.1.2 Data Transformation

Based on the principles of the reference model (section 2.3.4), we transform the raw data into case-by-variables tables. Although the content of the case-by-variables table depends on the specific task of the MCS analysis, we list the essential data with respect to the MCS analysis in Table 4.1 in order to determine the visual structure. The table has the following variables:

- Variable "Element" represents the element of an MCS dataset with the ID. An element may be a MCS or a basic event.

- Variable "Type of Element" is derived in order to distinguish the elements. A type may be "MCS" or "basic event".
- Variable "Relation" (referring to R1) has the type *Link* and represents the inclusion relations between MCSs and basic events (Figure 4.1). For an MCS, the value of this variable is a set of basic events, e.g., $MCS_4 = \{BE_2, BE_3, BE_5\}$ (Figure 4.1), and for a basic event, the value of the variable "Relation" is a tuple of MCSs that include this basic event, e.g., $BE_1 = \{MCS_1, MCS_3\}$.
- Variable "Failure Probability" (referring to R2, R3) is the native properties of the MCSs and basic events. We use the constant failure probability in our work.
- Variable "Unreliability level" is derived in order to qualitatively estimate the failure probability.
- A statistical variable, "Number of relations", is derived for the total links of each element. This may represent the order of an MCS or the number of occurrence of a basic event.

The failure propagation of MCSs (referring to R4) is a particular piece of data that depends on the structure of the CFT (Table 4.5). The data are independent on MCSs, so a visual structure needs to be designed for the failure propagation that can be effectively composed with the visualization of MCSs.

4.1.3 Visual Mapping

4.1.3.1 Design of the Layout

The relations between MCSs and basic events may be described as data consisting of two disjoint sets (i.e., a two-mode data) (Figure 4.1): an MCS set and a basic event set, whose relations can be represented as a bipartite graph, or a bipartite network graph. According to the definition of the graph $G = (V, E)$, MCSs and basic events are vertexes and the inclusion relations are edges. A network is usually represented using either the node-link diagram or the matrix layout [18]. As a final result, the 2D Cartesian matrix layout is selected. The decision process is depicted in the following paragraphs.

4.1.3.1.1 Benefits and shortcoming of the layouts. Basically, an important reason of selecting the matrix layout is that the rows and columns can naturally represent MCSs and basic events. The other reasons depend on the well-known benefits and shortcomings of the layouts:

- node-link diagram
  - benefit: intuitive representation of edges; less display space requirement.
  - shortcoming: overlapping of vertexes; crossing of edges.
- matrix layout
  - benefit: no overlapping of vertexes; no crossing of edges; orderability.
  - shortcoming: weak representation of the complex connections; large display space requirement.

The node-link diagram can display more elements in the same display space than the matrix layout can. However, along with the benefit, the visual clutter caused by crossing of links and node overlapping also appears. In some cases, the interaction "details-on-demand", which dynamically shows links of the selected nodes, may partly solve the issue. However, it also reduces the benefit of the overview of the relations between nodes.

Several studies compared the readability of the matrix layout and the node-link diagram using scientific experiments. Ghoniem et al. [69, 70] compared the layouts in different aspects, e.g., "node count", and "finding link". The evaluation also took the effects of different algorithms of the node-link diagram into account. To examine whether the above evaluation was also meaningful for complex graphs, Keller el al. [113] evaluated the matrix layout and the node-link diagram using the directed graphs with semantic structures, e.g., geographical location, instead of the random undirected graphs. The evaluations concluded that the matrix layout was better suited for large and dense graphs (>100 nodes) and the node-link diagram was more effective for small and sparse graphs. However, the node-link diagram was more effective for the task of "finding path" that was related to identifying a path between two nodes, regardless of whether the graph was large or small. The matrix layout was not good at representing the complex paths, particularly when identifying the indirect neighbors.

The evaluations provided evidence for determining the layout of our visualization. Because the MCS data may contain thousands of MCSs as well as basic events, representing large graphs is a significant point for our design. Since the density of graphs depends on MCSs generated for the specific systems, we do not take the density of graphs into account.

First, we consider the readability of the inclusion relations in the MCS data. The above evaluations revealed that the matrix layout was more effective for the task of "finding link" that is related to the identification of the relations between MCSs and basic events. The evaluations [22–24, 175] showed that the intrinsic orderability of the matrix layout can increase the readability of the quantitative properties and thus outperform the node-link diagram. Using the reorderable matrix, engineers may instantaneously rank the vertexes (i.e., MCSs or basic events) by the sequence of rows or columns.

Second, we consider the comparability of the quantitative properties of the MCS data. MCSs may be estimated and compared according to the quantitative variables, e.g., the failure probability. The edge-crossing of the node-link diagram reduces the comparability. The node degree (i.e., the number of the connected links) can intrinsically represent the order of the MCS and the number of occurrence of the basic event. In common application, the nodes with large degrees may be treated as the important elements. However, in the MCS analysis, the criterion is inverse for MCSs. A critical MCS usually has a relatively small order. This criterion represents a meaning that is against the visual perception of the node-link diagram. Additionally, the links of the unimportant nodes may cause the serious crossing issue and interfere with the identification of the links of the important nodes. Moreover, the degree of a node may also represent the number of occurrence of a basic event when the node represents a basic event. The larger the number of occurrence of a basic event is,

the important the basic event is. This criterion differs from that of MCSs. Thus, we cannot simultaneously assign the inverse semantic meanings to the node degree. Considering the reasons above described, we select the matrix layout rather than the node-link diagram.

4.1.3.2 Cartesian Coordinate vs. Radial Layout

We above discuss the matrix layout based on a Cartesian coordinate. The matrix layout can be alternatively represented using the radial coordinates.

Diehl et al. [31,54] investigated the readability of the matrix layout using Cartesian coordinates and radial coordinates. The studies considered not only the basic layouts but also the additional visual context and graphical properties that may increase the readability, e.g., color of border, in order to comprehensively compare the matrix layouts using different coordinates. The results showed that the Cartesian matrix layout brought both better accuracy and higher perception speed of the tasks with respect to the identification of positions. The authors suggested that the Cartesian matrix layout should be the first choice, unless there were clear reasons for using the radial matrix layout. The evaluations also estimated the readability of the two dimensions in each layout, i.e., row vs. column in the Cartesian matrix layout as well as sector vs. ring in the radial matrix layout. The results showed that there was no significant difference between the readability of row and that of column in the Cartesian layout. In contrast, in the radial layout, the sector significantly outperformed the ring. The cross-comparison showed the ranking of the readability of the four dimensions: $sector > row = column > ring$. The authors suggested that the Cartesian matrix layout had a better readability when the data of both dimensions were almost equally important, and the radial matrix layout was worth selecting when users only focused on one dimension.

We decide to apply the Cartesian matrix layout. The primary motivation is the better effectiveness of the Cartesian layout. In addition, the Cartesian layout provides almost the same readability for rows and columns. This property is beneficial for the MCS analysis because MCSs and basic events have equal significance in principle when analyzing their relations.

In short, according to the above-mentioned points, the 2D Cartesian matrix layout is finally determined as the basic visualization layout for the MCS data.

4.1.3.3 Complements of the Layout

Besides the relations between MCSs and basic events, there are still variables (Table 4.1), e.g., “Failure Probability”. These variables are significant to estimate MCSs and basic events. In order to represent these variables, we combine two additional tables with the matrix layout according to IDs by the composition principle [132] (Figure 4.2). In this way, the composed matrix layout has two additional columns (on the left) representing the failure probability and order of MCSs, with the additional rows (at the bottom) representing the failure probability and number of occurrence of basic events. An important benefit is that the matrix layout may have a more flexible orderability according to different variables.

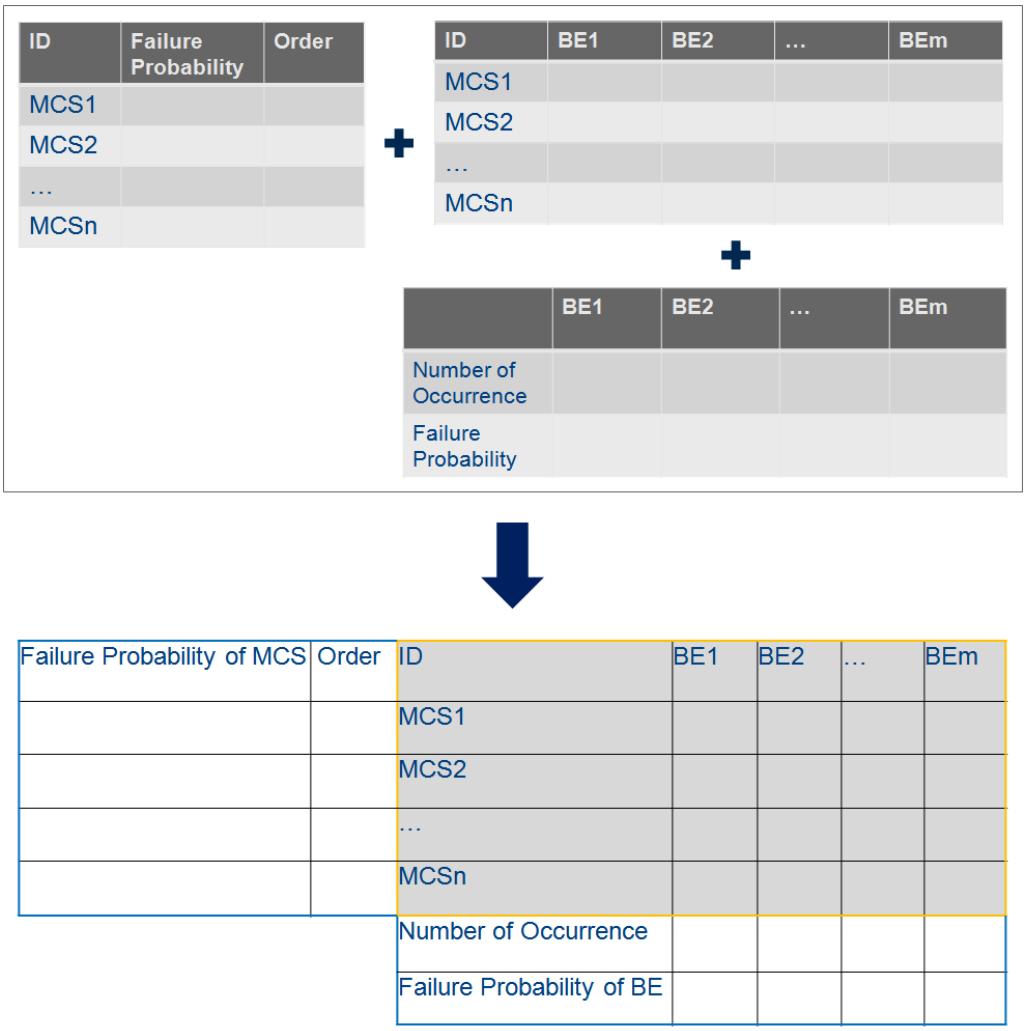


Figure 4.2: Composition of the matrix layout and the tables for properties. The left table is combined with the matrix according to the IDs of MCSs. The lower table is combined with the matrix according to the IDs of basic events.

Element	N	→ row/column
Type of Element	N	→ position
Relation	Link	→ filling
Failure Probability	Q	→ length
Unreliability level	O	→ color hue
Number of relations	Q	→ length

Table 4.2: Graphical properties for the MCS data.

4.1.3.4 Graphical Properties

we determine the graphical properties of the vari ables of the MCS data (Table 4.2). We use the "filling (or called density)" of cells to represent the variable "Relation". This is the commonly used method for representing the co-occurrence between the row and column in a matrix layout. Because bar graphs are suitable to embed in matrix cells, we use graphical property "bar length" to visualize the quantitative variables: "Failure Probability" and "Number of relation". In order to clearly identify the unreliability levels of MCSs and basic events, we use the color hue to encode this ordinal variable. The details of the graphical properties are described in Chapter 5.

4.1.3.5 Visualization of the Failure Propagation

In order to represent the failure propagation of MCSs, the CFT structure needs to be integrated in the matrix layout. Considering the the definition and convention of the CFT, we apply the Cartesian hierarchical network layout to the CFT structure. The design concept is similar to the design of the visualization of the CFT structure for the importance analysis argued in section 4.2.3.2. Because the CFT structure is complex and needs a lot of display space, it is not suitable to show the complete structure. We need to find a way to to dynamically show the CFT structure. The possible concepts are the Focus+Context and Overview+Detail. An evaluation [16, 45] showed that the Focus+Context was more effective than the Overview+Detail. In addition, the Focus+Context provides more continuous and integrated information. Thus, we use the Focus+context concept to enlarge the focused cell of the matrix layout, and the semantic zooming concept to show the corresponding parts of the CFT structure in the enlarged cell. The details of the interactions are introduced in section 5.1.7.

4.2 Regarding the Importance Analysis

4.2.1 Data

The importance analysis based on the CFT is aimed at estimating the risk contribution of each basic event to the top event. Because CFT components are modeled according to the hierarchical system architectural model, the critical CFT components directly correspond to the vulnerable system components. A CFT consists of two types of structures: the nesting relations between CFT components and the CFT

Basic Event	N	BE_1	BE_2	...	BE_n
Importance value	Q				

Table 4.3: The case-by-variables table for the importance of basic events.

Element	N	$Component_1$	...	$Component_n$	BE_1	...	BE_m
Containing relation	Link		...		ϕ	...	ϕ

Table 4.4: The case-by-variables table for the containing relations in the CFT.

structure (or logical structure) representing the data flow. Figure 2.3 shows an example that the sub-components (marked with colors) are nested in the system-level CFT, which are connected by a logical structure. Figure 2.4 shows a CFT having a multi-level nesting structure. The hierarchies of the nesting structure correspond to the hierarchies of the system architectural model in principle. Thus, the vulnerable system components can be located in the system model by means of identifying the corresponding critical CFT components in the nesting structure. The CFT structure is the basis for the analysis of the failure flow regarding the important basic events. According to the motivations introduced in section 3.1.2, the required data (shorter form "R") is summarized with respect to the following aspects:

- Estimation of basic events using the importance analysis.
  - R1: the importance of basic events.
- Locations of CFT components by considering the system architectural model.
  - R2: the inclusion relations between CFT components and basic events.
  - R3: the nesting relations between CFT components.
- Analysis of the failure flow of the important basic events.
  - R4: the structures of CFT components with failure probabilities.

4.2.2 Data Transformation

We transform the required data into case-by-variables tables. Table 4.3 focuses on the quantitative importance of basic events (referring to R1). We use the Fussell-Vesely importance measure [68] to calculate the importance of basic events. Table 4.4 describes the containing relations in the CFTs. Variable "Containing relation" represents that a CFT component may include basic events and other (sub-)CFT components. We use this variable for both required data "R2" and "R3" because the two relations may be combined and represented as a tree: the system-level CFT component is the root, sub-CFT components are non-leaf nodes, and basic events are leaves. The failure flow depends on the logical structures of CFT components (referring to R4) that is described in Table 4.5.

4.2.3 Visual Mapping

We first consider the analysis of the critical CFT components. This analysis depends on the distribution of the important basic events over CFT components. In this case,

CFT Element	N	BE_1	...	$Gate_1$	...	$In{-}port_1$	...	$Out{-}port_1$	...
Type of Element	N								
Failure Probability	Q								
Logical Connection	Link								

Table 4.5: The case-by-variables table for the CFT structure.

	Node-Link	Treemap	Sunburst	Icicle	Matrix	Radial Tree
Readability of hierarchical data	+	−	−	+	−	−
Space efficiency	−	+	+	+	−	−

Table 4.6: Summary of the benefits and shortcomings of hierarchical layouts.

the information of the nesting relation is more significant than that of the logical data flow. Thus, we decide to integrate the importance of basic events (Table 4.4) with the containing relation (Table 4.3). By taking the large-scale CFTs into account, we decide to design a compact view for the nesting relations in order to maintain a suitable overview of the distribution of the important basic events. We dynamically present the failure flow of the requested basic events in order to efficiently utilize the screen space. In this section, we first talk about the design of the visualization layouts for the nesting relations between CFT components, and then argue the visualization concepts for the failure flow; finally we discuss the layout composition strategies for the final spatial substrate.

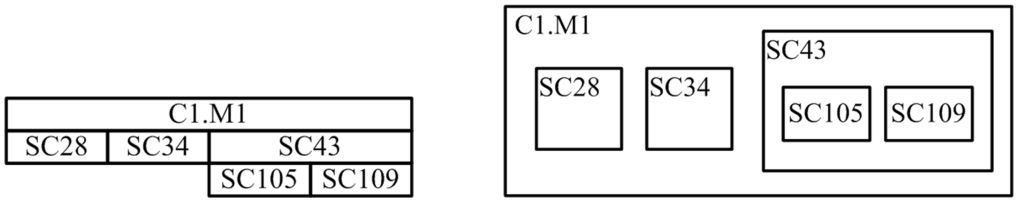
4.2.3.1 Layout of the Architectural View

The containing relations described in Table 4.4 can be interpreted as a tree structure. The importance value is a quantitative attribute attached to basic events. We design an *architectural view* associating the importance of basic events with the containing relations between CFT components. We determine the basic layout of the architecture view by taking three aspects into account: readability of hierarchical data, space efficiency, and composition of layouts. Basically, the main categories of representations for trees are the node-link diagram, the treemap layout, the sunburst layout, and the icicle diagram. Additionally, we also consider two particular layouts: the matrix-based visualization and the radial tree.

We first consider the commonly used layouts: the node-link diagram, the treemap, the sunburst layout, and the icicle diagram. Barlow et al. [13] evaluated the readability of these layouts with respect to the ease of interpretation, comparison of leaf sizes, as well as the participants' preference. The results showed that the icicle diagram and the node-link diagram were more readable for 2D hierarchical structures. There were few differences between the icicle diagram and the node-link diagram regarding readability.

We then consider the space efficiency of the layouts. McGuffin et al. [134] performed a mathematical evaluation for the space efficiency of 2D tree representations. The authors proposed a set of metrics in regard to the representations of nodes and labels. The results showed that the compact layouts, i.e., the treemap, the icicle diagram, and the sunburst layout, had similar space efficiency that was much better than the space efficiency of the node-link diagram.

We then consider the matrix layout and the radial tree. Previous evaluations



(a) Icicle Diagram

(b) Diagram for the nesting relations between CFT components

Figure 4.3: Concept of the icicle-diagram-style architectural view.

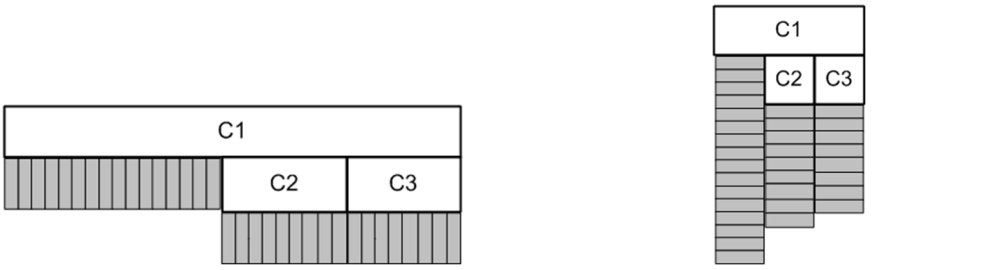
[69,70,113] concluded that the task of "path finding" cannot be effectively performed using the matrix layout. However, our analysis tasks are just strongly related to the identification of paths in trees. A path between the root (i.e., the system-level CFT component) and a specific node (i.e., a specific sub-component) represents the location of this component in the nesting structure of the CFT. Additionally, the space requirement of the matrix layout is even larger than that of the node-link diagram. Thus, the matrix layout is not suitable. The radial tree has an unfavorable space efficiency [134] and worse readability [32] than the Cartesian tree has.

The benefits and shortcomings of the layouts are briefly summarized in Table 4.6 with respect to the readability of hierarchical data and space efficiency. The icicle diagram is the best choice that can not only effectively represent hierarchical relations, but also efficiently use screen space for visualizing large data.

An important consideration of our design is the feasibility of the layout composition between the architectural view and the CFT structure. We discuss this point in section 4.2.3.4. The results show that the icicle diagram is more suitable than other layouts regarding the layout composition.

Finally, as a result, we apply the icicle diagram to the architecture view (Figure 4.3). Rectangles of the architectural view represent CFT components. Hierarchies of the rectangles represent the nesting relations between CFT components. The architectural view may be used to represent not only the system-level CFT component, but also the sub-CFT components. The architectural view of a CFT component may be treated as the combination of the architectural views of the sub-CFT components (Figure 4.5 (a)).

4.2.3.1.1 Leaf nodes. We represent basic events (i.e., leaf nodes of the tree structure) as small rectangles under the rectangles for their parent CFT components. We apply the variant of the icicle diagram, called iceray diagram [157], where the orientation of the list of leaf-rectangles is perpendicular to the parent rectangle (Figure 4.4 (b)). By the icicle concept, the diagram may be too long to effectively analyze when there are thousands of basic events. By the iceray concept, the length of a rectangle depends on the number of its sub-CFT components that is usually much smaller than the count of the directly and indirectly included basic events. On the other hand, the leaf node lists under different parent rectangles may share the vertical space, so that the height of the iceray diagram usually does not increase very quickly. Additionally, the iceray diagram is favorable for distinguishing basic events



(a) Icicle tree: horizontal leaf node list (b) Iceray tree: vertical leaf node list

Figure 4.4: The icicle concept and the iceray concept.

and CFT components because their rectangles are listed in different orientations.

An important property of the icicle diagram is the proportional length of rectangles according to the specific attribute of the leaf nodes. However, the proportional relations between rectangles play an important role in the importance analysis, neither with respect to the quantitative importance value nor regarding the number of basic events. Moreover, losing this property in the iceray diagram does not lead to information loss in the importance analysis.

4.2.3.2 Representation of the Logical Failure Flow

The representation of failure flow depends on the CFT structure that is treated as a DAG network. We need a visualization layout that is able to effectively represent complex paths of the failure flow and logical relations between basic events. We had two alternative concepts:

- directly represent failure flow in the architectural view using additional links as ArcTrees [144] did.
- individually represent failure flow and compose the representation with the architectural view.

The first concept needs to integrate the failure flow into the architectural view by adding necessary elements and links, e.g., gates. This may either strongly change the structure of the architectural view or add new elements in a crowded space, usually under the corresponding rectangles. In the case of changing the structure of the architectural view, the analysis based on the nesting relations between CFT components may be interrupted. By another concept, in a crowded space, it would be difficult for us to apply a global unidirectional orientation to these new added elements like the orientation of the ordinary structure of CFT components. This hampers the analysis of failure flow along the complex critical paths and the analysis of the logical relations between the important basic events. In addition, the new added links may cause serious visual clutter on the architectural view, particularly when presenting the paths across different hierarchies. Using this representation concept, we can neither represent the intuitive failure flow, particularly for multiple important basic events, nor maintain a satisfactory view for the containing relations between CFT components. The individual representation can provide a clear CFT structure for representing failure flow without hindering the architectural view.

	Architecture (upper)	CFT (upper)
Architecture (lower)	case 1 (Fig. 4.5 (a), (b))	case 3 (Fig. 4.5 (e))
CFT (lower)	case 2 (Fig. 4.5 (c), (d))	case 4 (Fig. 4.5 (f))

Table 4.7: The layout composition cases. Combination between architecture view and CFT structure with respect to the view position: upper part and lower part. Figure 4.5 shows examples of the combination cases.

In short, although the integrated representation may represent the failure flow in a space-efficient manner, the topology of the critical path is not intuitive. In contrast, the individual representation is more intuitive, but this requires more space. Because our goal is to effectively analyze the complex failure flow, the intuition is more meaningful for us. Thus, we decide to individually visualize the failure flow and compose this visualization with the architectural view. The logical relations between basic events are meaningful in the context of the analysis of failure flow. For this reason, we present the whole logical CFT structure of the requested CFT component rather than the only part related to the failure flow. Additionally, this helps engineers to estimate the importance of basic events when analyzing the structures of the specific CFT components.

We need to determine the basic visualization layout for the CFT structure. The DAG is commonly represented using the node-link diagram and the matrix-based visualization [18]. The characteristics of both concepts are briefly described in section 4.1.3.1.1. The previous evaluations [69,70,113] compared the readability of both layouts. The results showed that the node-link diagram was more effective for representing the complex paths in network graphs. For this reason we prefer to use the node-link diagram in order to guarantee the intuitive critical path of the failure flow.

Nodes of a network graph can be placed according to either the semantic meaning or the structural rules. First, we consider the semantic meaning depending on the attribute of nodes. In our work, the most important attribute is the importance of basic events. Section 4.2.3.5 argues that a common graphical property is needed for the importance values in both the architectural view and the CFT structure. The position of node is obviously inappropriate to use as the common graphical property. Thus, we arrange nodes according to the CFT structure rather than semantic meaning.

The CFT structure has a global unidirectional orientation from basic events to the top event. Layering nodes may facilitate the analysis of the critical path of failure flow. Thus, we use a hierarchical network [15,53,133,183] for the CFT structure. We also consider whether the radial layout is suitable. The study [32] that evaluated the readability of the Cartesian layout and the radial layout of hierarchical data. The results showed that the traditional Cartesian layout was more readable. In addition, we apply the top-to-bottom direction to the Cartesian node-link diagram by taking the convention of the CFT analysis into account.

4.2.3.3 Layout Composition

We need a suitable way to compose the node-link CFT structure and the architectural view. Zhao et al. [212] talked about all possibilities of the layout composition

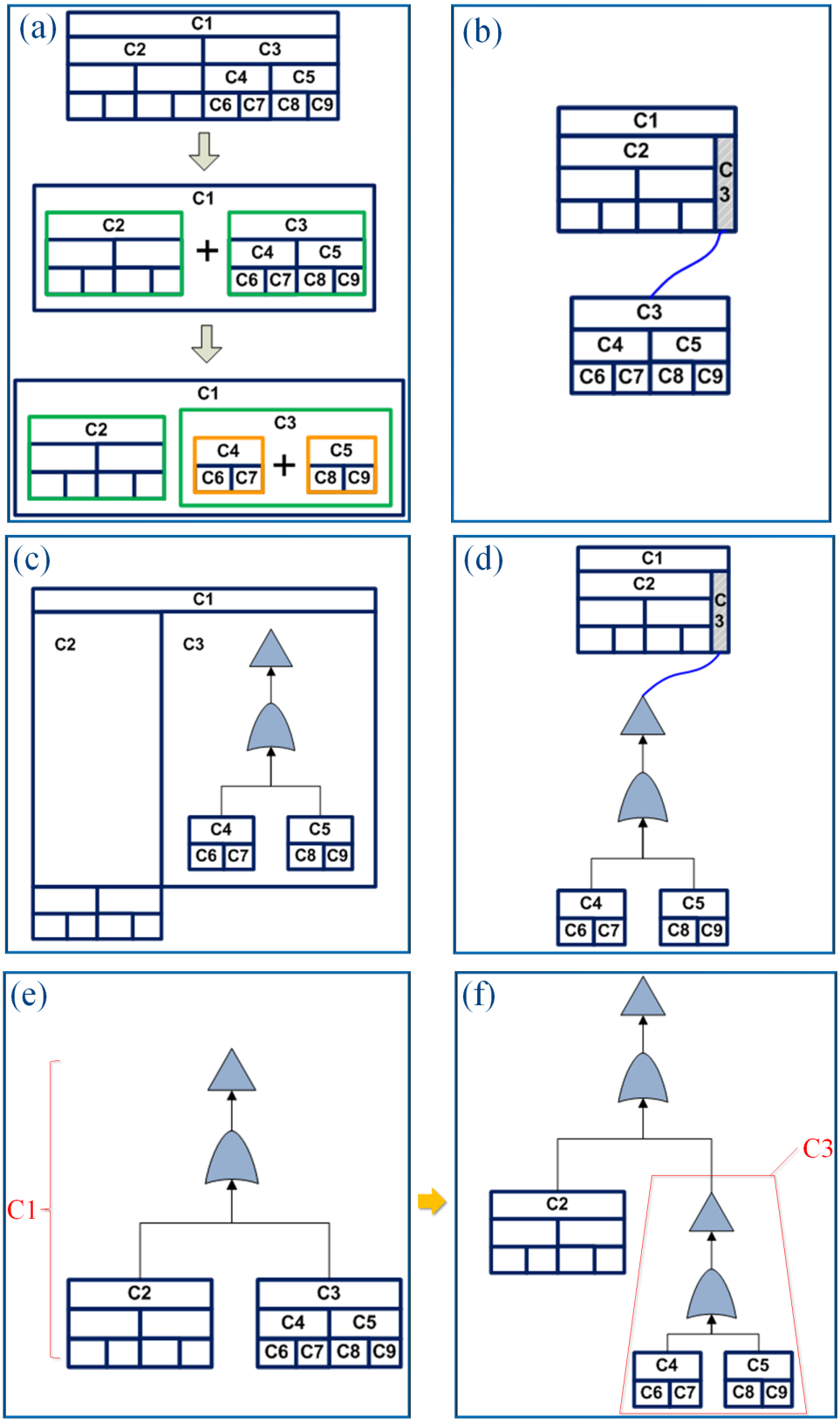


Figure 4.5: Cases of the layout composition between the architectural view and the logical CFT structure. (a) Architectural view of the CFT component "C1" that is a combination of the (sub-)architectural views of the sub-components. (b) Showing the architectural view of "C3" outside the main architectural view. (c) Showing the structure of "C3" with the inside composition strategy. (d) Showing the structure of "C3" with the outside composition strategy. (e) Showing the structure of the top-level component "C1". (f) Showing the structure of "C3" when the structure of "C1" has been shown (i.e., (e) is the previous state of (f)).

between the node-link diagram and the treemap based on a defined taxonomy. Rufiange et al. [165] also argued a taxonomy of the compound visualization layouts and discussed the composition between the matrix layout and the node-link diagram.

We consider all possible cases of the combination between the architectural view and the CFT structure (Table 4.7) in order to comprehensively analyze the composition. Although we focus on the combination between the architectural view and the CFT structure, we still list the particular cases in the table: combination of different (sub-)architectural views, and combination of different node-link CFT structures. Both cases play important roles for the layout composition. Figure 4.5 shows the conceptual examples based on the composition cases. To easily understand the concepts, we do not represent basic events (i.e., leaf nodes) in the examples. The concepts are available in practical applications when considering the basic events. We discuss the possible composition cases summarized in Table 4.7 and select suitable composition methods according to the requirements of the importance analysis.

Case 1 describes that an architectural view of a CFT component can be treated as the combinations of the architectural views of the sub-components. For example, in Figure 4.5 (a), the main architectural view of "C1" includes the (sub-)architectural views for the sub-components: "C2" and "C3". The architectural view of "C3" includes the views of the sub-components "C4" and "C5". An alternative composition is described in Figure 4.5 (b) that pulls the view of "C3" out of the parent architectural view. This alternative concept is not appropriate for our visualization because the composed view does not provide any new information for the importance analysis, however, this takes additional space.

Case 2 may appear when analyzing the logical structure of the sub-CFT components. For example, when showing the logical structure of "C3" in Figure 4.5 (a), the architectural views of the sub-components "C4" and "C5" are embedded into the logical structure of "C3" (Figure 4.5 (c) and (d)). This way, engineers are allowed to investigate the data of the sub-components and the deeper nested sub-sub-components ("C6", "C7", "C8", and "C9") while maintaining the structure of "C3". Referring to [165, 212], we have two alternative strategies for composing the architectural view and the CFT structure: the inside composition strategy (Figure 4.5 (c)) and the outside composition strategy (Figure 4.5 (d)).

Using the inside composition, the CFT structure is shown inside the enlarged rectangles of the architectural view, whereas using the outside composition, the CFT structure is shown outside and linked to the architectural view. Regarding the icicle diagram, the inside composition is not suitable because enlarging a rectangle will cause a height increase of the whole hierarchy. When displaying CFT structures in multiple hierarchies, the height of the architectural view may greatly increase. Simultaneously, the length of the icicle diagram may also greatly increase. In this case, the readability and the space efficiency may be strongly reduced. In contrast, the outside composition (Figure 4.5 (d)) adapts well to the icicle diagram. When showing the logical structure of "C3", a small rectangular indicator is drawn in the architectural view to replace the part representing the component "C3" (Figure 4.5 (d)). Without this replacement, the data of the CFT component "C3" (i.e., the sub-component "C4" and "C5") in the architectural view of "C1" and those in the logical structure of "C3" are duplicate. Using this concept, the confusion of the duplicate

data is addressed and the repeatedly occupied display space can be recycled.

Case 3 may occur when showing the logical structure of the top-level component "C1", the structure of "C1" directly replaces the previous architectural view (Figure 4.5 (e)).

Case 4 describes the combination between logical structures of different CFT components. In Figure 4.5 (e), the logical structure of "C1" has been displayed. Based on this structure, we then display the logical structure of the component "C3" in Figure 4.5 (f). Both of the node-link CFT structures are merged into a whole.

4.2.3.4 Discussing alternative Layouts of the Architectural View with respect to Layout Composition

We also consider the feasibility of the layout composition between the CFT structure and the alternative layouts of the architectural view, i.e., the treemap, the sunburst layout, and the node-link diagram. The node-link diagram is first rejected because there will be confusion when using the node-link diagram to represent both the containing relations and the logical failure flow. The inside composition strategy is not possible for the treemap and the sunburst layout because of the aspect ratio problems.

We then discuss the outside composition strategy with respect to the matter of space recycling for the architectural views represented using the treemap and the sunburst layout. For displaying the logical structure of a sub-component when we recycle the corresponding rectangle (for treemap) or sector (for sunburst) in the parent architectural view, the aspect ratios (of the treemap and the sunburst layout) need to be recalculated and the structure of this architectural view may greatly change. This may lead to confusion between the new view and the old view. If we do not recycle the space, our visualization could be very space-inefficient. A deeper-nested sub-component may repeatedly occupy space when showing the CFT structures of the direct or indirect parent components in the manner of Figure 4.5 (c). Additionally, the algorithms of the treemap need an initial rectangle as an input. It is difficult to determine an appropriate initial rectangle for the architectural view of a sub-component, in which the shapes of rectangles are the same as those in the parent architectural view. In short, the treemap and the sunburst layouts are not suitable to combine with the node-link CFT structure.

4.2.3.5 Importance Bar

We need a common graphical property for the importance of basic events in order that engineers may compare the importance of basic events both in the architectural view and in the CFT structure. According to the preference of graphical properties (Figure 2.27), length is a suitable graphical property to represent the quantitative data. We propose a rectangular graphical property called the *importance bar* (Figure 6.1) for representing the basic event with the importance value. The importance bar has a fixed border and a color-filled part encoding the importance value. We embed importance bars into the CFT structure below the basic event nodes. In the architectural view, we use the importance bars to replace the initial leaf-rectangles. This

way, the global comparison of the importance is possible regardless of whether the importance values are represented in the architectural view or in the CFT structure.

4.3 Regarding the Safety Improvement Process

4.3.1 Data

According to the motivations (section 3.1.3), the goal of the visualization of the safety improvement process is to visually facilitate the identification of the improvement solution(s) consisting of a series of design modifications. The improvement process concentrates on two phases: construction of solutions and analysis of solutions.

4.3.1.1 Data with respect to the Construction of Solutions

The three main steps in each iteration for the construction of solutions require the following raw data (shorter form "R"):

- Step 1: Identify the important basic events.
  - R1: the importance of basic events for the identification.
- Step 2: Apply and test the risk reduction hypothesis by system modifications.
  - R2: the type of modification: substitution or redundancy.
  - R3: the value of modification: change of failure probability of the initial basic event.
  - R4: the cost of modification.
- Step 3: Evaluate results of risk reduction.
  - R5: the failure probability of the top event of the updated CFT model. This is used to decide whether the reduced system risk is acceptable.
  - R6: the impact on the top event by design modification. This is used to identify the optimal modification that causes the most risk reduction (section 2.1.6.1).
  - R7: the cost-effectiveness of the design modification. This is used to identify the optimal modification that is the most cost-effective (section 2.1.6.1).
  - R8: the gap between the updated failure probability of the top event and the goal value. This is important context that shows how much failure probability still needs to be reduced.

The safety improvement process involves much different data that cannot be simultaneously visually presented. We focus on the data that directly influence the identification of the modification (R1, R5, R6, R7, R8), and the data that are essential for analyzing the existing modifications (R2, R3, R4).

Modification	N	$Modification_0$	$Modi._1$	...	$Modi._n$
Connection	Link				
Basic event	N				
Modification type	N				
Modification value	Q				
Modification cost	Q				
Cost-Effectiveness	Q				
New failure probability of TE	Q				
Single impact on TE	Q				
Goal of the improvement	Q				

Table 4.8: Data table for design modifications.

4.3.1.2 Data with respect to the Analysis of Solutions

The analysis of existing solutions is aimed at reviewing the related data and analyzing the statistical information of the data, e.g., the total cost. A solution is completely constructed when it reduces the failure probability of the top event to a given acceptable value. We focus on the reduction up to the goal value rather than the exhaustive risk reduction. The overloaded reduction may lead to a large improvement; however, it may simultaneously also generate a high cost. In our work, we assume that any completely constructed solution reduces the initial risk to the same goal value. In this case, all complete solutions have the identical effect of risk reduction. Thus, when identifying the most cost-effective solution, we analyze the total cost of each solution instead of the total cost-effectiveness.

4.3.2 Data Transformation

We transform the raw data into case-by-variables tables: the table for the importance (Table 4.3) (referring to R1), and the table for design modifications (Table 4.8) with the following variables:

- Variable “Modification” represents the design modification with the ID.
- Variable “Connection” represents the connection between modifications.
- Variable “Basic event” indicates the basic event related to the modification.
- Variable “Modification type” (referring to R2) describes the type of the modification.
- Variable “Modification value” (referring to R3) represents the reduction of failure probability of the addressed basic event. Depending on this variable, engineers may investigate the improvement of the vulnerability being addressed.
- Variable “Modification cost” (referring to R4) represents the quantity consumed for the modification, e.g., money, time, and human resources. The type of the cost needs to be determined at the beginning of the safety improvement process. It is an important piece of information for optimizing modifications (section 2.1.6.1) as well as the meaningful context for analyzing solutions.
- Variable “New failure probability of TE” (referring to R5) describes the new risk state of the updated CFT model with respect to the failure probability

Solution	N	$Solution_1$	$Solution_2$	...	$Solution_n$
Included Modification	N				
Total cost	Q				
Total steps	Q				

Table 4.9: The case-by-variables table for data of solutions.

of the top event. According to the principle described in section 4.3.1.2, when the system risk becomes acceptable by a modification, we assign the given goal value rather than the actual value to variable "New failure probability of TE" of the modification.

- Variable "Single impact on TE" (referring to R6) represents the impact on the top event by a single modification without considering the summed influence of the previous modifications.
- Variable "Cost-Effectiveness" (referring to R7) measures the modifications by taking the balance between cost and the single impact on TE into account. We define the cost-effectiveness as the ratio between the reduction of the failure probability of the top event and the cost of the corresponding modification.
- Variable "Goal of the improvement" (referring to R8) is used for estimating the gap between the actual risk and the goal value.

FT structure provides meaningful information for the safety improvement process. This supports the understanding of the causal dependencies between failure mechanisms and the way a failure propagates through the system. We apply CFT to our visualization instead of the ordinary FT because CFT components correspond to the components of the system architectural model. This supports the identification of the critical (physical) parts in the system model corresponding to the important basic events. In addition, the CFT structure supports to analyze the effect of modifications along the way a failure propagates through the system when reviewing solutions. Thus, we associate the CFT structure with the visual safety improvement process. The data of the CFT structure are described in Table 4.5.

We transform the data of existing solutions into Table 4.9. For visualizing a solution, the included modifications are essential data. In order to support the identification of the optimal solutions, we additionally derive the statistical data from modifications: total cost, and total steps.

4.3.3 Visual Mapping for Solution Construction

The results of the safety improvement process, i.e., modifications, are sequentially connected according to the order of the identifications. The connection has a tree structure and is usually treated as a decision tree. For this reason, we treat a decision tree that organizes the modifications as the basic structure of our visualization. Because the process starts at a non-modified system design, we add a root node to the decision tree in order to represent the initial system design (Figure 2.10: root is on the left side). Other variables of the modification are visualized and integrate with the decision tree node. The decision tree needs to be interactively constructed during the safety improvement process. As a result, we finally design a risk-reduction plot

(Figure 7.2 (2)) to integrate the variables of the design modification with the decision tree. The following sections depict the design concepts of the spatial substrate illustrated in Figure 4.6.

4.3.3.1 Decision of Layout

There are various layouts for representing the tree structure. We discuss the node-link diagram, the treemap layout, the sunburst layout, the icicle diagram, and the matrix layout.

The evaluations [69, 70, 113] proved that the matrix view was not suitable to represent complex paths. Barlow et al. [13] evaluated the readability of the treemap layout, the sunburst layout, the node-link diagram, and the icicle diagram. The results showed that the node-link diagram and the icicle diagram were the most favorable. We still consider whether the layouts are suitable to represent attributes of nodes, i.e., properties of modifications. The node-link diagram is more suitable because this layout has sufficient space to visualize and integrate multiple attributes of nodes. In contrast, the compact layouts, i.e., the icicle diagram, the treemap, and the sunburst layout do not have good potential for the integration. The aspect ratio issue may be caused when using the compact layouts and the nodes deeper on the tree do not have sufficient space for integrating the attributes [194]. In addition, we discuss the node placement in section 4.3.3.1.1 in order to represent the important information of modifications using the node position. Thus, we prefer to use the node-link diagram to represent the decision tree rather than the compact layouts. The node-link diagram is the basic layout of our visualization.

We still need to argue the Cartesian coordinates and the radial layout. A study [32] evaluated the Cartesian traditional tree layout, the Cartesian orthogonal tree layout, and the radial layout for the node-link diagram. The authors compared these layouts in terms of finding common ancestors, the exploration behavior, and the effect of tree orientation. The results showed that the Cartesian layout was more preferred with respect to readability while the radial layout was more space-efficient. The study also argued that the top-to-bottom orientation was more effective than the other alternatives. In our work, the top-to-bottom direction can perceptually represent the reduction process of the failure probability. For those reasons, we apply a top-to-bottom Cartesian node-link diagram to the decision tree.

4.3.3.1.1 Node placement. The structure of the node-link diagram depend on either the semantic rules or the structural rules [182]. The most significant data of a design modification are related to two aspects: the cause (i.e., the important basic event) and the effect (i.e., impact on top event). The corresponding data are the "Basic event" and "New failure probability of TE" (Table 4.8). We propose a 2D scatter plot, called the *risk-reduction plot*, where the x-axis has a nominal scale of basic events, and the y-axis represents the failure probability of the top event from the initial value to the goal value of the safety improvement (Figure 4.6). The nodes of the decision tree, i.e., modifications, are placed in the plot.

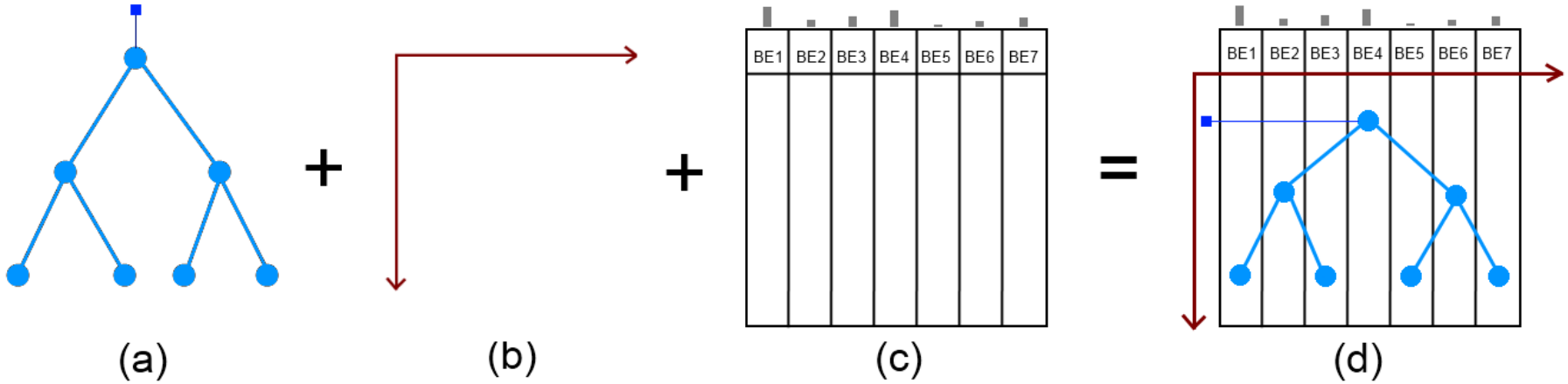


Figure 4.6: Risk-reduction plot. (a) Decision tree representing the sequential design modifications. The small blue box is the root of the decision tree that represents the initial system design. (b) Scatter plot. The x-axis represents a list of basic events and the y-axis represents the failure probability of the top event. (c) List of basic events with the bars representing the importance of basic events. (d) Composed risk-reduction plot.

Modification	N	→ point
Connection	Link	→ line
Basic event	N	→ position
Importance of basic event	Q	→ length
Modification type	N	→ shape
Modification value	Q	→ length
Modification cost	Q	→ length
New failure probability of TE	Q	→ position
Single impact on TE	Q	→ length
Cost-Effectiveness	Q	→ length
Goal of the improvement	Q	→ position

Table 4.10: Graphical properties for the safety improvement process.

4.3.3.2 Association with the Importance Analysis

The importance of the basic event is a significant piece of data for the safety improvement process. Because the construction of the decision tree is a dynamic iterative process, in each iteration the updated importance of basic events needs to be compared. In this case, the overview of the importance of all basic events is required. We decide to integrate this data with the risk-reduction plot. Because the x-axis of the risk-reduction plot represents all of the basic events, this is a suitable basis for integrating the importance values. Bars are appropriate for representing the quantitative importance values and suitable to be integrated with the basic event list on the top of the risk-reduction plot (Figure 4.6). This way, in each iteration of the safety improvement process, engineers may start at the identification of the important basic event(s) by analyzing the bars.

4.3.3.3 Summary of Graphical Properties

Besides the visualized data above mentioned, we also design graphical properties for other data relevant to the improvement process. We visualize the data in Table 4.10 and compose these visualizations using a risk-state node (Figure 7.1) that provides a compact representation for the modification in the risk-reduction plot.

In this way, the overview of the sequential connections of modifications and the visualized essential data of each individual modification are combined by the risk-reduction plot. Thus, engineers may focus on the details of the specific modifications while maintaining an overview of the relations between modifications. This allows engineers to intuitively analyze interesting modifications and explore patterns. For example, answering the questions: which modification generates the highest cost in a specific solution. The details of the visualization properties and interactions are introduced in Chapter 7.

4.3.4 Association with CFT Structure

Due to the significant help of the CFT structure for the safety improvement process, we associate the enhanced CFT structure for the importance analysis with the risk-reduction plot by the horizontal positions of basic events that appear in both views (Figure 7.2 (1)). In this way, the logical relations between the basic events

corresponding to modifications can be treated as context information for reviewing the improvement process.

4.3.5 Visual Mapping for Solution Review

For analyzing the existing solutions, we decide to build additional plots associated with the risk-reduction plot (Figure 7.2 (3)). The plots provide clear views for the trend of the solutions by representing the statistical data of solutions. These views support to analyze the existing solutions and identify the optimal ones.

Chapter 5

Visualization of Minimal Cut Sets

Based on the design concepts discussed in section 4.1, we propose a matrix-based visualization approach called *MCS Matrix* [209] in order to facilitate the analysis of the MCSs of the CFT.

5.1 MCS Matrix

5.1.1 Matrix View

The core of the MCS Matrix is a matrix view that is compound with additional columns and rows. The central matrix view (Figure 5.1, area (1)) represents the relations between MCSs and basic events, where rows represent MCSs and columns represent basic events. A cell at the intersection of a row and a column is filled when an MCS of the row includes the basic event of the column, else the cell is empty. For estimating the basic events, cells are filled with colors that encode the unreliability levels of basic events. The schema of the colors is introduced in section 5.1.2. In this way, a color-filled cell represents two points: the existence of the relation between an MCS and a basic event, and the unreliability level of the basic event.

The additional columns on the left side (Figure 5.1, area (2)) represent the failure probability and the order of the MCSs. We use the bar graph to represent the failure probability. The larger the value is, the longer the bar is. In order to draw readable bars for the very small probabilistic values, we provide a configurable scale for the bar length instead of the initial scale between 0 and 1. The lower and upper bounds of the scale are defined as the minimal and maximal failure probabilities of the MCSs. In this case, the bar represents a relative measure rather than the absolute value. We provide an alternative logarithmic scale for the bar because the linear scale is not suitable to represent the large exponential interval, e.g., [1e-10, 1e-5]. The bar graph increases the amount of the displayed rows on a limited screen because it takes up less display space than textual representations.

In many cases, the qualitative estimation for the failure probability provides a way to quickly assess MCSs. Bars for the failure probability are assigned colors according to the unreliability levels of MCSs. Using the color-filled bars, engineers may quickly estimate the criticality of MCSs while quantitatively comparing the MCSs according to the failure probability. Similar to the bars for failure probability,

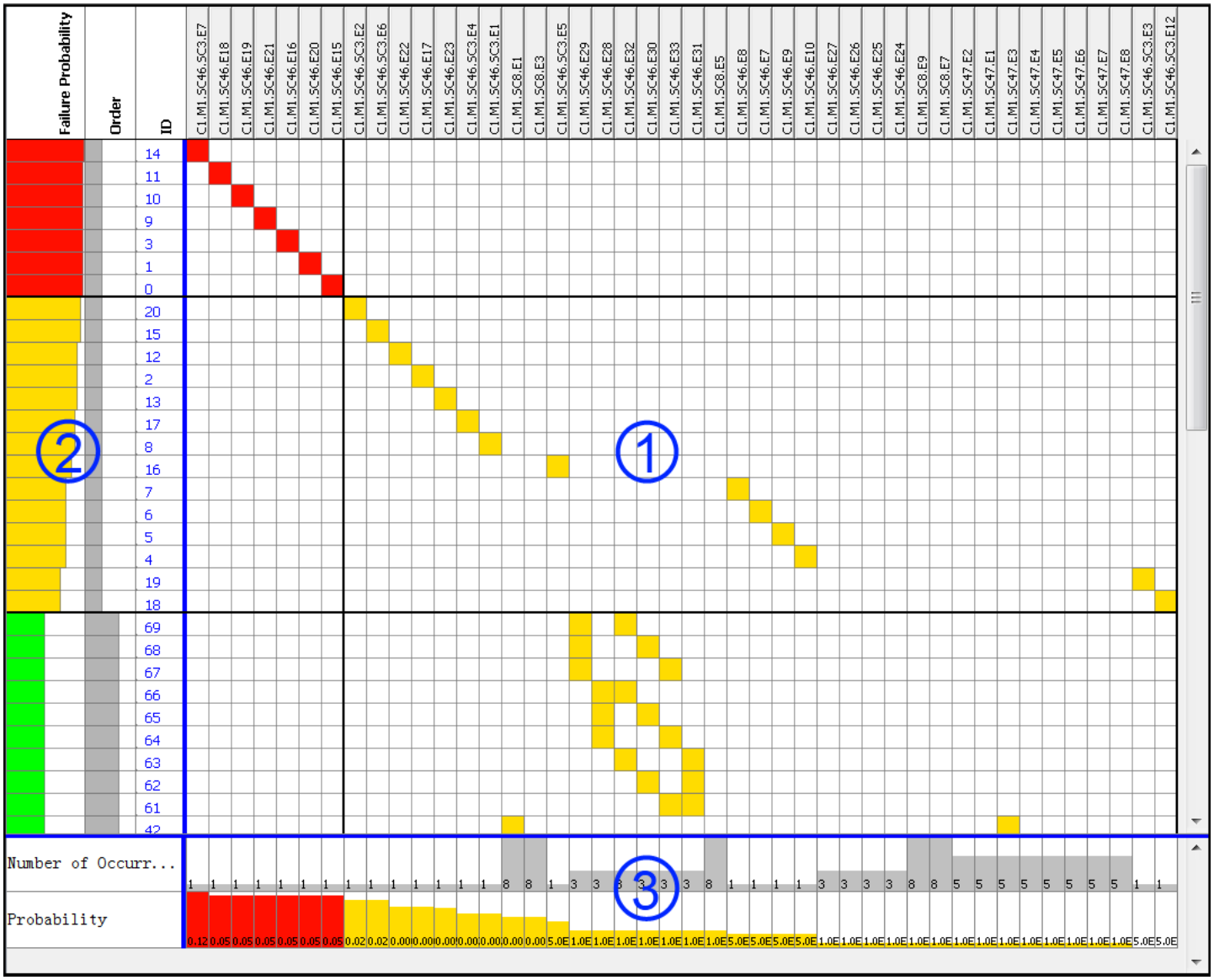


Figure 5.1: Matrix View. (1) Relations between MCSs (rows) and basic events (column). (2) Property area of MCSs (including the failure probability, the order, and the ID). (3) Property area of basic events (including number of occurrence, and the failure probability).

the bars representing the order of the MCSs are also proposed. The smaller the order of an MCS is, the shorter the bar is, i.e., the more critical the MCS is. The gray color of the bars is only used for visibility and does not provide any semantic meaning.

For the additional rows on the bottom (Figure 5.1, area (3)), we also use bars to represent the failure probability of the basic events and the number of occurrences of basic events. The unreliability levels of basic events are represented using colors of the bars. Colors of the bars for the number of occurrences do not have any meaning.

5.1.2 Unreliability Levels

In many cases, engineers purpose to quickly estimate the failure probabilities of the MCSs according to the risk acceptance. This estimation may facilitate the identification of the critical MCSs. A commonly used qualitative rating method is the traffic-light rating system [2] that classifies data into three classes and uses colors to encode the classes. We apply this method to classifying the unreliability (i.e., failure probability) into three levels based on the specified thresholds, and assign colors to the levels:

- critical level (red): the failure probability is extreme high. The failure needs to be addressed as soon as possible.
- moderate level (yellow): the failure probability is still higher than a commonly accepted value. The failure should be addressed but not urgently.
- acceptable level (green): the failure probability is in a safe range and may be neglected.

The unreliability level concept is also available for basic events. There may be respective configurations (i.e., threshold and colors) of the unreliability level for MCSs and basic events. When applying the same configuration to basic events and MCSs, engineers may analyze the quantitative relations between an MCS and its basic events, e.g., a critical basic event (red) and a moderate basic event (yellow) may together lead to a failure scenario (i.e., MCS) that has a moderate level probability (yellow). The values of the "cause" (i.e., basic events), and the value of the "effect" (i.e., the MCS), can be intuitively investigated.

5.1.3 The Grouping Methods

Without appropriately indexing, engineers have to manually identify the important MCSs by looking through all rows. If there are a large number of MCSs, this task will require a great deal of effort. We provide the grouping methods respectively for rows and columns by using the orderability of the matrix layout.

5.1.3.1 Grouping of Rows

We introduce the grouping methods by means of depicting the grouping by the failure probability of MCSs. First, the MCS rows are sorted by the failure probability in descending order. The MCS rows are naturally aggregated based on the unreliability levels. Each resulting aggregation is treated as a *group*. There is a one-to-one

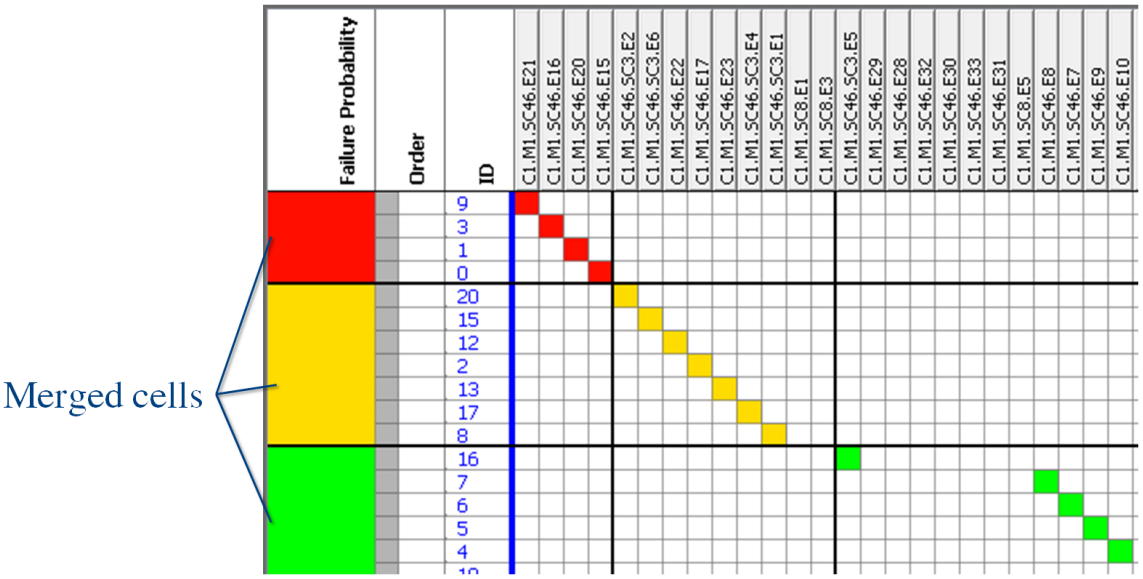


Figure 5.2: Merged cells in the first column are used as indicators of groups.

mapping between groups and unreliability levels. Colors are treated as indicators of the groups. Then, the column representing the failure probability is moved to the first position in order to announce that MCS rows are sorted by the failure probability (Figure 5.1, in area 2). The series of manipulations is called *grouping*; more precisely, the grouping by failure probability. In case engineers much more focus on the criticality of MCSs rather than the quantitative estimation, they are allowed to merge the cells in the column of failure probability based on the colors (Figure 5.2). The merged cells can provide the better indication for MCS groups without confusion caused by the length of bars. The grouping by the order of MCSs is basically the same. In simple terms, MCS rows are sorted by the orders of MCSs in ascending order, and then the relevant column is moved to the first place.

MCS Matrix provides a second-level sorting for the grouping method. This is particularly useful for the analysis of the order of MCSs. Figure 5.3 shows an example. Rows are primarily grouped by the order of MCSs. In each group, MCSs have the same order. In this case, the second-level sorting prioritizes the MCSs inside each group. As a result, the MCSs of the first group (with the order of 1) are clearly classified into three sub-groups (Figure 5.3 area (1)). The green sub-group may be neglected because the probability of occurrence is very low, even though the MCSs having the order of 1 are usually critical.

5.1.3.2 Grouping of Columns

We also provide grouping methods for columns in order to prioritize the basic events. Similar to the grouping of MCS rows, we propose the grouping methods for columns according to either the failure probability of basic events or the number of the occurrence of basic events. Figure 5.1 shows a view by grouping failure probability for both rows and columns.

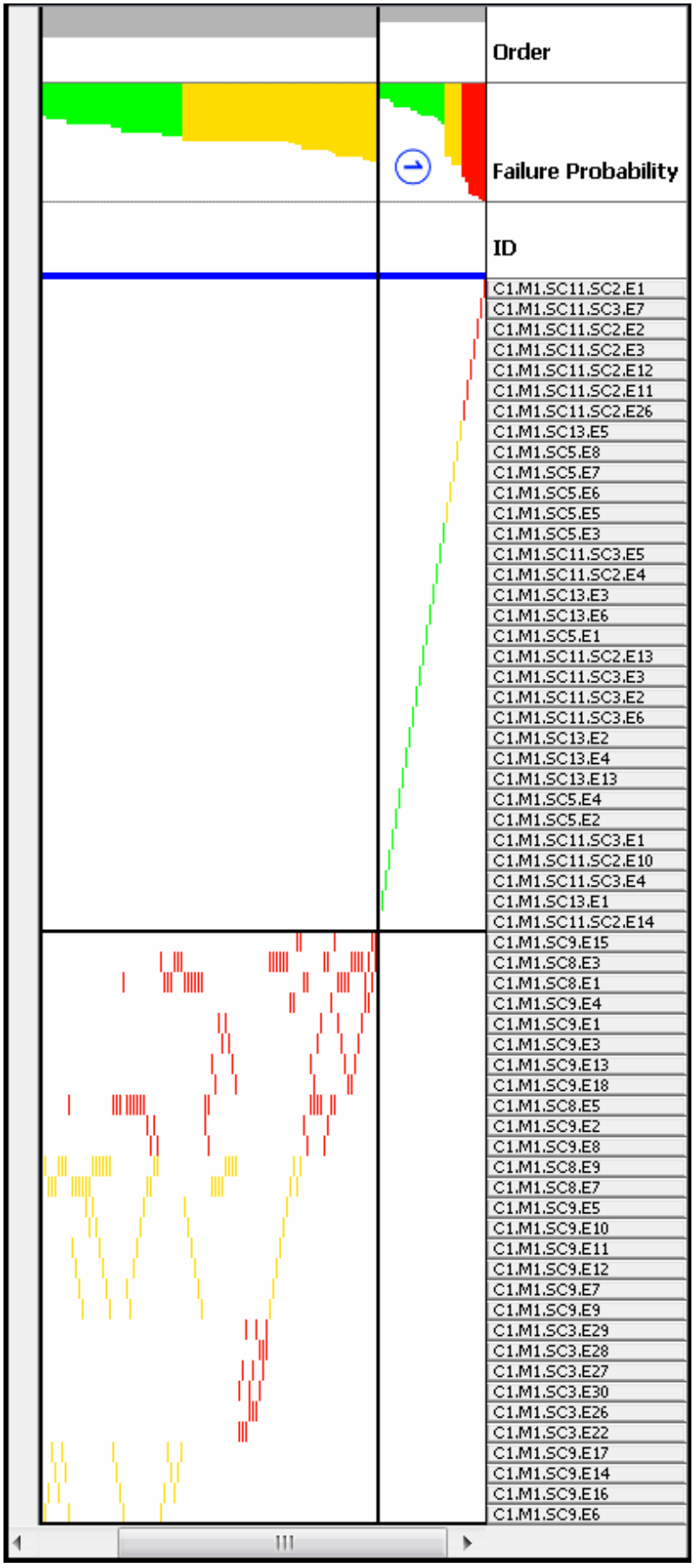


Figure 5.3: Second-level sorting for MCS groups (891 MCSs in total). MCS rows are primarily sorted by the orders of MCSs. Then the rows of each group are sorted by the failure probability.

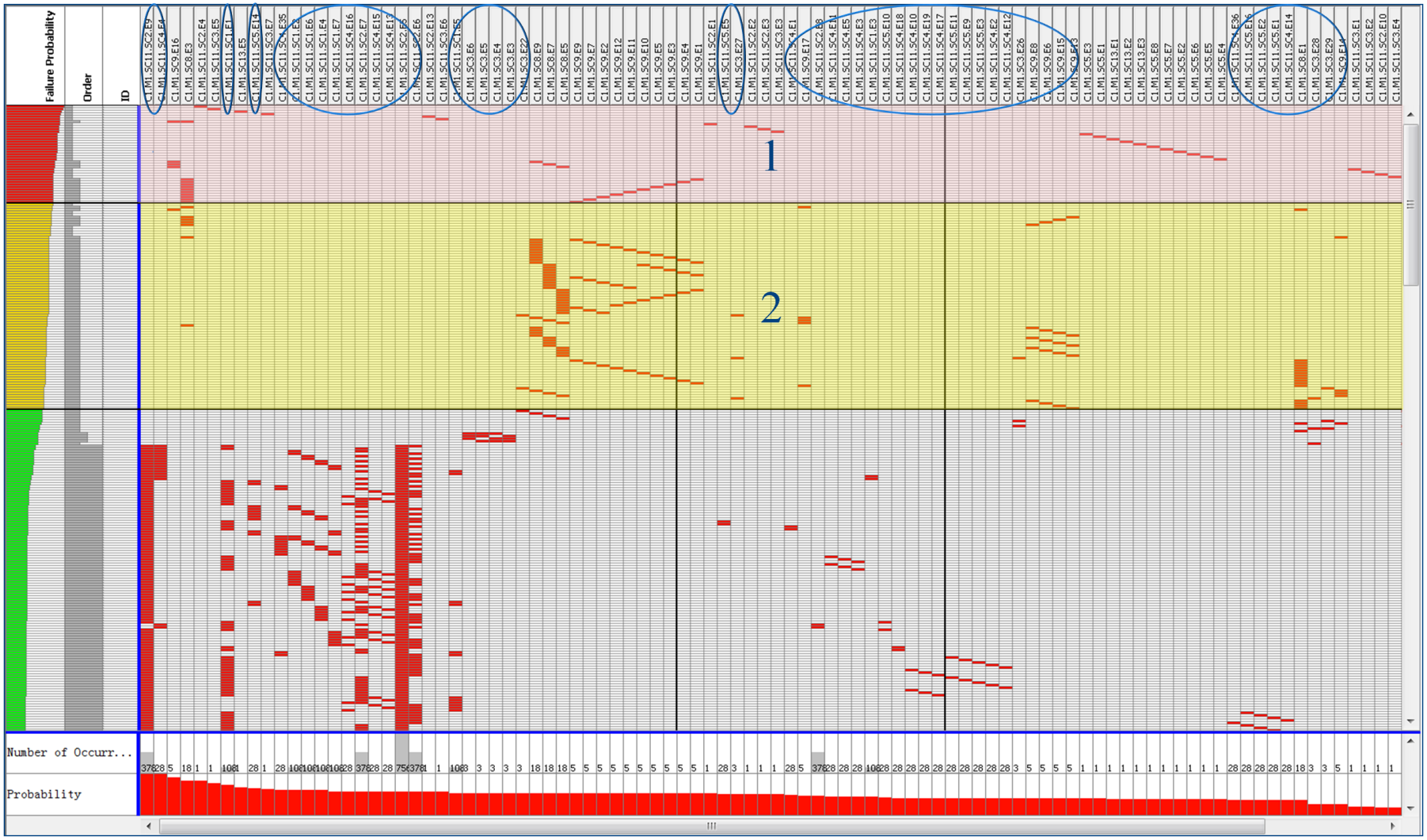


Figure 5.4: Columns are grouped by the failure probability (891 MCSs and 104 BEs in total). (Colors of the areas 1 and 2 are manually marked in the image.)

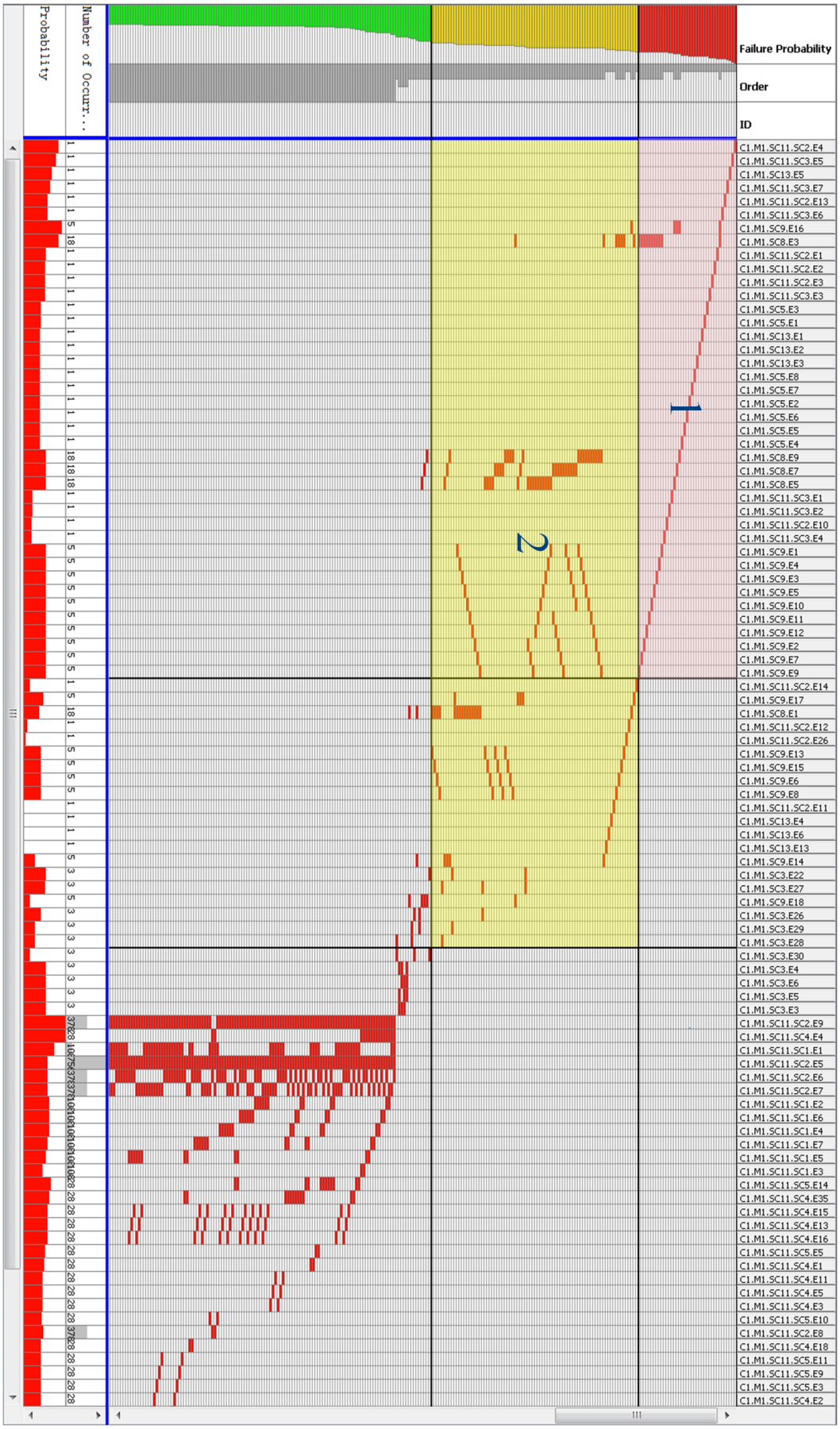


Figure 5.5: Columns are sorted by using the associated ordering concept (891 MCSs and 104 BEs in total). The columns related to the critical MCS rows are preferentially displayed. (Colors of the areas 1 and 2 are manually marked in the image.)

5.1.3.2.1 Associated ordering. In many cases, engineers care much more about the basic events related to the critical MCSs. The columns representing the basic events included by the critical MCSs may be sparsely distributed by using the normal ordering, e.g., columns are grouped by the failure probability in Figure 5.4, and the columns related to the critical MCS (i.e., red rows) are sparsely distributed because of the columns (marked by the blue circles) that are not related to the critical MCSs. In this case, these columns take up a lot of display space, but do not provide meaningful information for the critical MCSs. Thus, we need to find a way to preferentially display the meaningful columns on the screen.

To address this issue, we propose an associated ordering concept for columns that takes the criticality of MCSs into account. First, the columns are sorted according to the criticality of the related MCSs rather than the criticality of the basic events. If a basic event is related to multiple MCSs, we only consider the highest criticality of the MCSs. It guarantees that the columns related to the important MCSs can be preferentially displayed. If two columns have identical priority, columns are again sorted by the failure probability of the basic events.

Using the associated concept, columns may be more rationally displayed. In addition, the pattern may be identified. Figure 5.4 shows that columns are sorted by the failure probability of basic events. In order to analyze the basic events related to the critical MCSs, we need to go through all of the columns (area (1)). Figure 5.5 shows a view by using the associated ordering. The columns related to the critical MCS are aggregated in area (1). The area that must be concentrated on for analyzing the critical MCSs becomes smaller.

In a similar way, the columns related to the moderated MCS rows are distributed in area (2). The area in Figure 5.5 is smaller than that in Figure 5.4. The relations (i.e., filled cells) are aggregated in the corresponding areas that are arranged like a stairstepping. The more important the MCS group is, the smaller the corresponding area is. In this way, when engineers analyze an MCS group, they only have to focus on the specific area corresponding to the MCS group rather than the whole matrix. In Figure 5.5, area (1) may be treated as the most important area because this shows the relations between the critical MCSs and the critical basic events. The associated ordering reduces the effort required for investigating the basic events related to the critical or moderate MCSs because of the smaller interesting areas.

5.1.4 Integration of Textual Data

Although engineers may quickly estimate the probability values by colors and bars, in many cases, the exact textual values still play an important role. The textual value may be used for confirming the estimation based on the graphical properties. Thus, MCS Matrix allows textual data to be displayed within the matrix cells. To avoid the visual conflict between the text and the bar in a cell, we provide an alternative method to separately display the bar in the upper part and the text in the lower part of the cell. For the failure probability of the MCSs, the text value is an important complement of the graphical representations. There are different purposes: colors indicate the criticality of the MCSs for a quick estimation; bars only provide the relative measurement for the comparison; texts provide the exact

values for confirming the above estimations.

5.1.5 Scaling

An overview of MCSs provides context information when focusing on the specific MCSs or basic events. In addition, it is helpful for pattern analysis that facilitates engineers to analyze the general safety situation of a system. For this reason, a method representing a satisfactory overview of a large number of MCSs is needed. On the other hand, detailed information is also significant, e.g., the exact failure probability values of MCSs. The ID is a piece of particular information that may not provide semantic information, but is very useful for navigating MCSs and basic events. The detailed information is the meaningful complement for the overview, and vice versa. Thus, effectively presenting detailed information in a satisfactory overview of MCSs becomes a new goal. We provide flexible scaling concepts for this purpose by applying the basic concept of the table lens technique that is introduced in section 2.4.1.1.

5.1.5.1 Uniform Scaling

In order to present as many MCSs as possible on a limited screen, we provide an *uniform scaling* for the matrix view. Row height may be uniformly reduced in order to adapt to the display space. Row height has the minimal value of 1 pixel (Figure 5.3). When displaying textual data in cells, the text font can be automatically adjusted to fit the size of cells. The uniform scaling strongly increases the visible rows in a limited screen. On a 22-inch monitor with a resolution of 1680x1050, the matrix view can effectively display a maximum of 700 rows with a height of 1 pixel. Similar to rows, engineers are allowed to uniformly reduce the column widths, too. With the help of uniform scaling, the matrix view may present a satisfactory overview of MCSs with the associated information.

5.1.5.2 Individual Scaling

Uniform scaling generates a satisfactory overview by reducing the geometrical size of rows and columns. However, some significant information is missing. The bars and textual values of MCSs are barely readable when row heights are greatly reduced. In contrast, showing this information may take up a lot of display space. A way to balance the detailed information with the overview is needed.

Our concept is to primarily maintain a satisfactory overview without showing detailed information until it is requested. We provide a table-lens-like interaction called *individual scaling* (Figure 5.6) that shows the detailed information including text and bars when mouse-clicking or hovering over rows. The row height has a default value (default: 16 pixels) that is large enough to display the text and graphical representations. Similar to rows, in order to clearly display the textual data of basic events, we also provide the individual scaling concept for columns. By using the individual scaling, engineers can effectively investigate detailed information while maintaining a satisfactory overview.

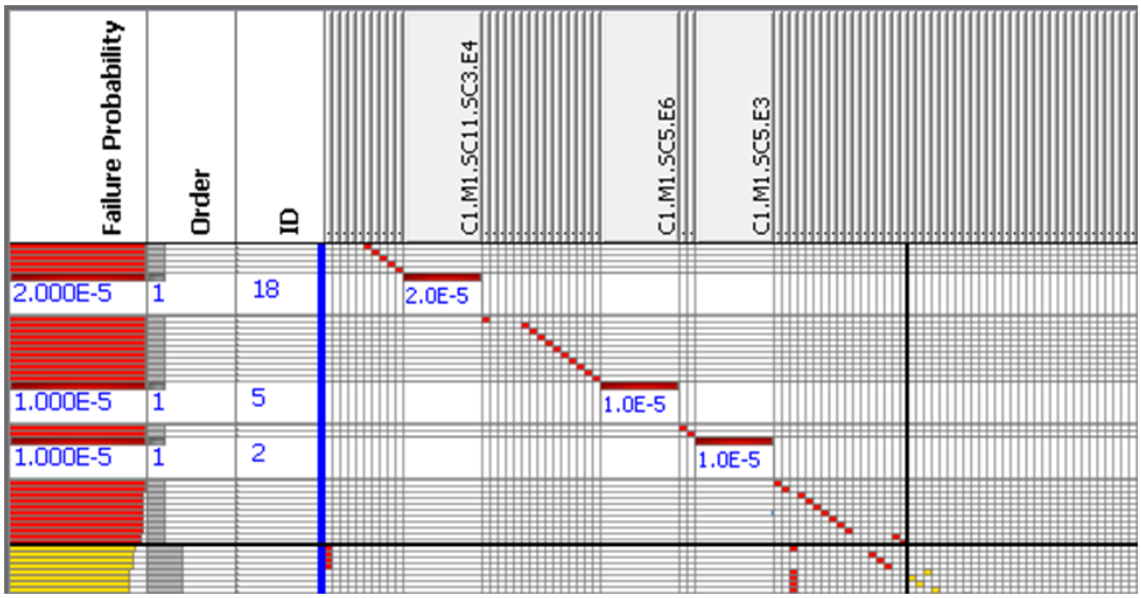


Figure 5.6: Individual scaling. This concept provides the appropriate width and height to the desired cells for clearly representing the detailed information.

5.1.5.3 Scaling by Groups

The more important an MCS is, the more valuable the information of the MCS is. Thus, it needs to display as much information of the important MCSs as possible in a limited screen while maintaining a satisfactory overview of MCSs. We propose an interaction called *scaling by groups* applying the DOI concept to the matrix view. The level-of-detail (LOD) and the DOI function need to be defined. In terms of the failure probability, the unreliability levels corresponds with the LOD:

- $LOD(Rows\ of\ the\ critical\ level) = 0$. Both graphical and textual representations are displayed in order to quickly and exactly investigate the associated data of MCSs.
- $LOD(Rows\ of\ the\ moderate\ level) = 1$. Graphical representations are clearly shown in order to quickly assess and precisely navigate the rows.
- $LOD(Rows\ of\ the\ acceptable\ level) = 2$. It shows only an overview of MCSs as context by applying a very small row height.

The DOI function describes how to render the visitable items according to the LOD. We define a discrete DOI function for row height rather than the commonly used continuous geographical changes.

- If LOD = 0, row height is 16 pixels. It is sufficient to show both graphical and textual representations.
- If LOD = 1, row height is 5 pixels. It guarantees that graphical representations are shown clearly.
- If LOD = 2, row height is 1 pixel. It strongly reduces the display requirement for only showing a rough overview as a context.

For example, in Figure 5.7, MCSs are scaled by groups according to the failure probability. The graphical and the textual information of the critical MCSs is clearly

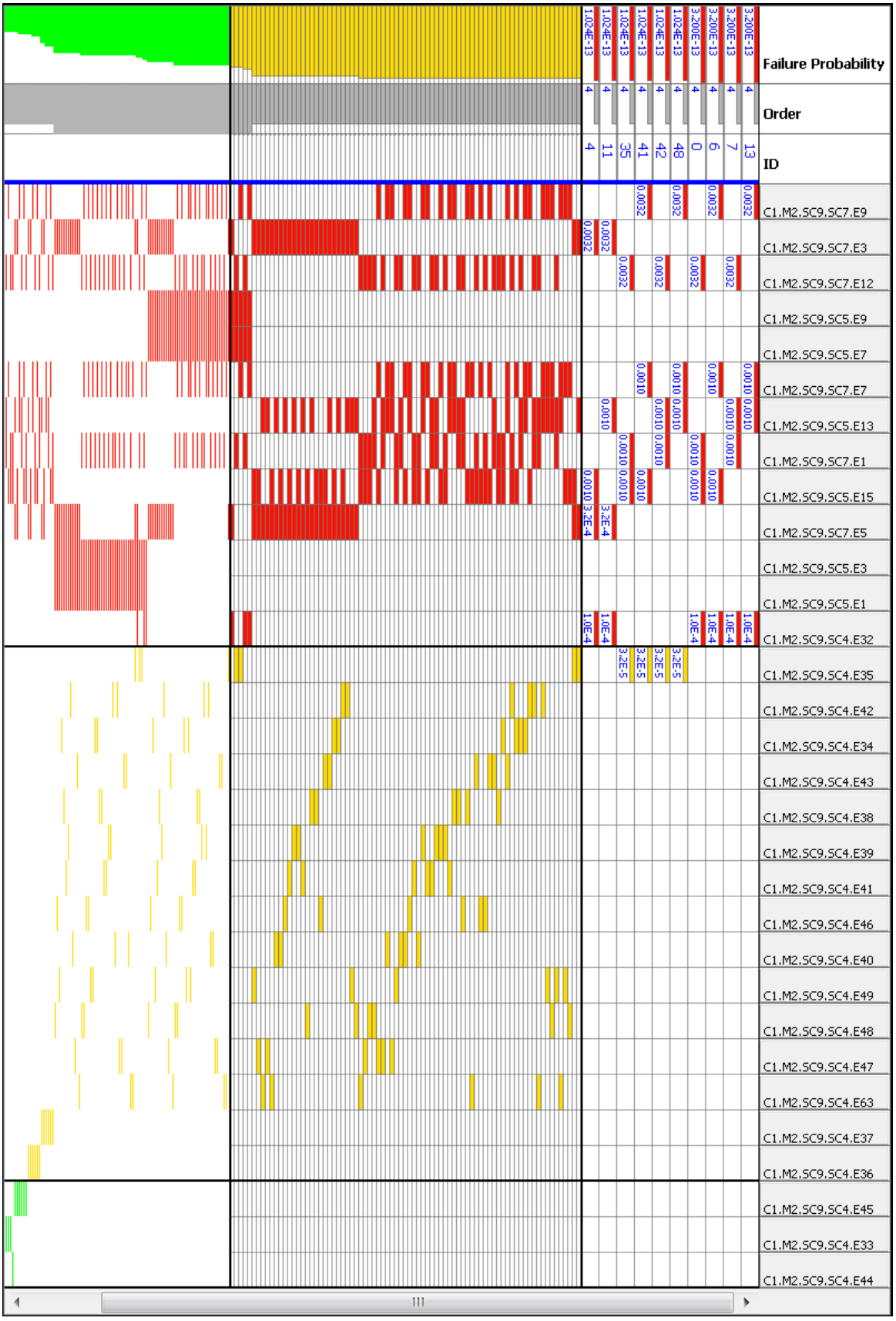


Figure 5.7: Scaling by groups (228 MCSs and 31 basic events in total). The red rows for the critical MCSs show both the graphical and textual representations. The yellow rows for the moderate MCSs only show the clear graphical representation without texts. The green rows for the acceptable MCSs only show a rough overview of the MCSs.

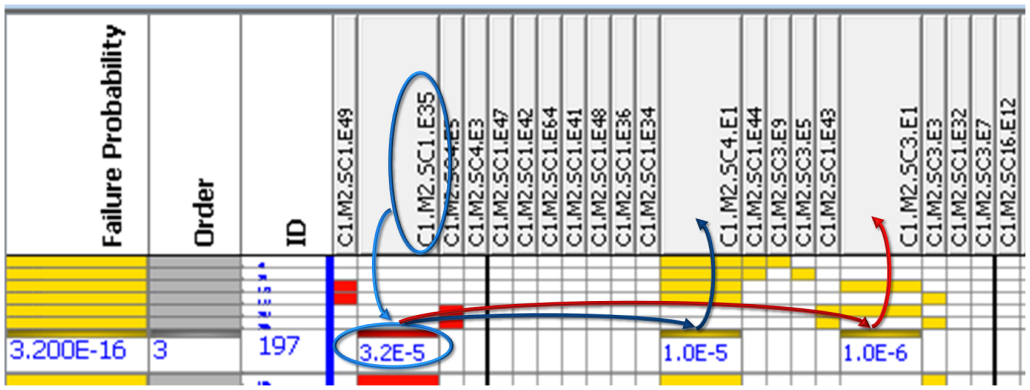


Figure 5.8: Identification of the partner basic events by analyzing the row of the common MCS.

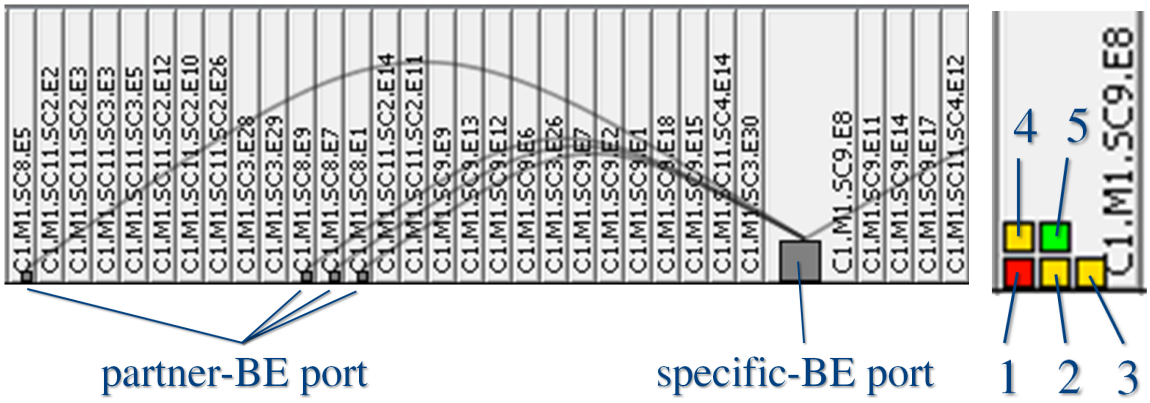
identified. The graphical representations of the moderate MCSs are able to be clearly read, but the textual values are not visible. For the acceptable MCSs, there is only an overview where engineers can just roughly investigate the pattern of these MCSs. Using the scaling by groups, engineers may allocate enough display space for the valuable data and effectively reduce display space of the less important information. This way, engineers can focus on the valuable information while maintaining an overview of the unimportant information.

5.1.6 Representing Relations between Basic Events

The analysis with respect to the specific basic event is significant in many application scenarios [136]. A significant case is to analyze a specific basic event that has a high failure probability and is difficult to improve. In order to reduce the likelihood of the occurrence of the failure scenarios that are caused by this basic event, engineers need to intensively address the "partner basic events" that may cooperate with this basic event to lead to the occurrence of the top event. For example, for a given MCS = {E1, E2}, if E1 has a high failure probability and is difficult to address, the partner basic event "E2" needs to be addressed in order to reduce the failure probability of the MCS.

The relations between basic events depend on the commonly related MCSs. For a specific basic event, engineers need to identify the partner basic events and analyze the causes of the relations. The identification process may be described as a path: specific basic event → MCSs → partner basic events. The matrix view can represent the path by going through the color-filled cells. Figure 5.8 shows an example of identification of the partner basic events by analyzing the row of the common MCS. The specific critical basic event is "C1.M2.SC1.E35" (marked with a circle). We analyze the MCS "197" that includes this basic event in order to identify the partner basic events. Along the row of the MCS, two partner basic events are identified. Along the columns we get the IDs of the partner basic events in corresponding header cells. However, when the amount of the related MCSs is large, it is difficult to effectively identify the partner basic events.

Thus we decided to find a way to represent the relations between basic events.



(a) The specific-BE port and the partner-BE ports.

(b) The MCS ports.

Figure 5.9: Relation ports.

Edges were integrated with specific layouts in previous studies (section 2.4.1.2) in order to represent the additional relations. The edges clearly represented the specific relations between the objects that were represented by other visualization layouts. We provide rectangular *relation ports* to represent the specific basic events, their partner basic events, and the common MCSs. We propose a large-sized port called a *specific-BE port* to represent the specific basic event and a small-sized port called a *partner-BE ports* to represent the partner basic event (Figure 5.9 (a) and Figure 5.10 (a)). A specific-BE port is connected with one or more partner basic event ports by curved lines in order to represent the relations between the specific basic event and its partner basic events. The connected relation ports forms a relational graph centering on the specific basic event. Multiple specific-BE ports are allowed to simultaneously represent different specific basic events (Figure 5.10 (b)).

The partner basic events may be by prioritized taking the criticality of MCSs into account. Referring to the path "specific basic event → MCSs → partner basic events" when a common MCS related to the specific basic event is critical, the partner basic events included by this MCS need to be preferentially addressed. We propose a medium-sized port called an *MCS port* to represent the related MCS (Figure 5.9 (b) and Figure 5.10 (c)). The color of a MCS port represents the unreliability levels of the MCSs. When there are many MCS ports for a specific basic event, we need to effectively arrange these ports in order to adapt to the limited display space. We horizontally present these ports from left to right (Figure 5.9 (b), the number is the order of MCS ports). If one line does not fully display these ports, we use multiple lines that are arranged from bottom to top. In this case, the list of these ports starts at the lower left corner of a header cell of the matrix view, and ends at the top right corner. In short, the arrangement of the MCS ports ranks the MCS ports according to their failure probabilities and preferentially presents the important MCS ports in a limited header cell. Using the MCS ports, engineers may prioritize the partner basic events according to the criticality of the MCS ports. To reduce the visual clutter, the MCS ports appear on only one header cell at a time, and the curved lines are translucent. In addition, because ports cannot present more details of the

MCSs, we show the detailed information of the corresponding MCS rows in the matrix view (Figure 5.10 (b) and (c)).

5.1.7 Integration with CFT Structures

In order to analyze the failure propagation with respect to MCSs, engineers investigate the failure flow of basic events along the CFT structure. Based on the design concept (section 4.1.3.5), MCS Matrix integrates logical structures of CFTs into the matrix view using the focus+context technique in combination with semantic zooming concept. CFT components are used as black boxes and a basic event only directly works inside its parent CFT component. For this reason, engineers usually focus more on the failure flow of a specific basic event within the bounds of the direct parent CFT component.

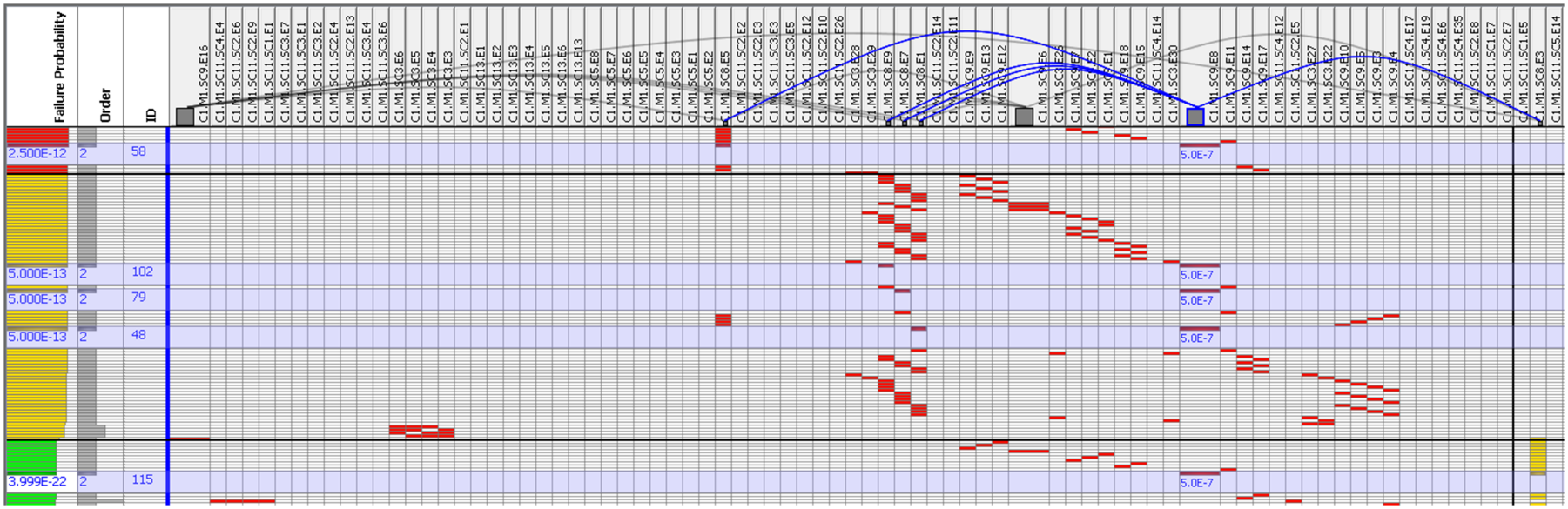
Based on this consideration, when a color-filled matrix cell is double-clicked, the cell is enlarged and displays an embedded view. The embedded view shows the internal logical structure of the direct parent CFT component of the basic event corresponding to the enlarged cell (Figure 5.11). Regarding the space requirement, only one embedded view can be shown at a time. In the embedded view, the critical path representing the failure flow is highlighted using blue. Nodes are assigned colors according to their unreliability levels. The nodes along the critical path are filled with colors, whereas other nodes only have a colored border: the basic events included by the MCS of the current row (but not of the current column) have a thick border and the rest of the nodes are marked with a thin border. In this way, engineers are able to focus on the failure flow while maintaining the overall MCSs as the context.

In addition, we provide a global path view on the left side of the matrix in the current row. There are two motivations for applying this view. First, the embedded view can provide the critical path of only one basic event at a time, so it still needs an overview of the critical paths of all basic events included by a specific MCS. Second, the embedded view only provides a local path within one CFT component. It still needs to show the paths over the whole CFT. This global path view is a good complement for the embedded view.

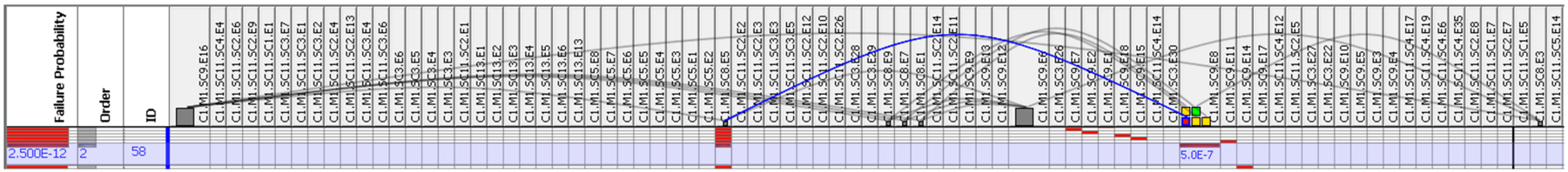
The global path view presents the overall critical paths of the basic events included by a specific MCS (Figure 5.12). Each node has the same shape and color as those in the CFT structure. In order to reduce the complexity of the global path, we directly link the nodes between different CFT components without using in-/outports. Considering the identification of CFT components, we use translucent gray blobs to indicate the scope of CFT components, so that the nesting of CFT components can be clearly represented. It should be noted that the CFT component of the system level (i.e., the top-level component) does not have a blob. Figure 5.12 shows an example that the global path view includes a CFT component of the system level (no blob), a sub-component, and three sub-sub-components.

The detailed information of the nodes of the global path view is dynamically displayed with a tooltip. Similar to the CFT structure shown in the embedded view, the global critical paths of the specific basic event are dynamically highlighted in blue. We provide interaction for the cooperation between the embedded view and the global path view. When selecting a basic event node in the global path view, the

(a) Relations with respect to single specific basic event.



(b) Relations with respect to multiple specific basic events. When selecting a basic event, the relations with respect to the basic event are highlighted. Detailed data of the MCSs including the selected basic event are presented in the matrix view.



(c) MCS relations with respect to a specific basic event. The port related to a critical MCS is selected and the highlighted curve indicates the partner basic event “C1.M1.SC8.E5”. The matrix view shows that the corresponding MCS has the ID of “58” and the failure probability of “2.5E-12”.

Figure 5.10: Connected relation ports. Curved lines represent the relations between basic events.

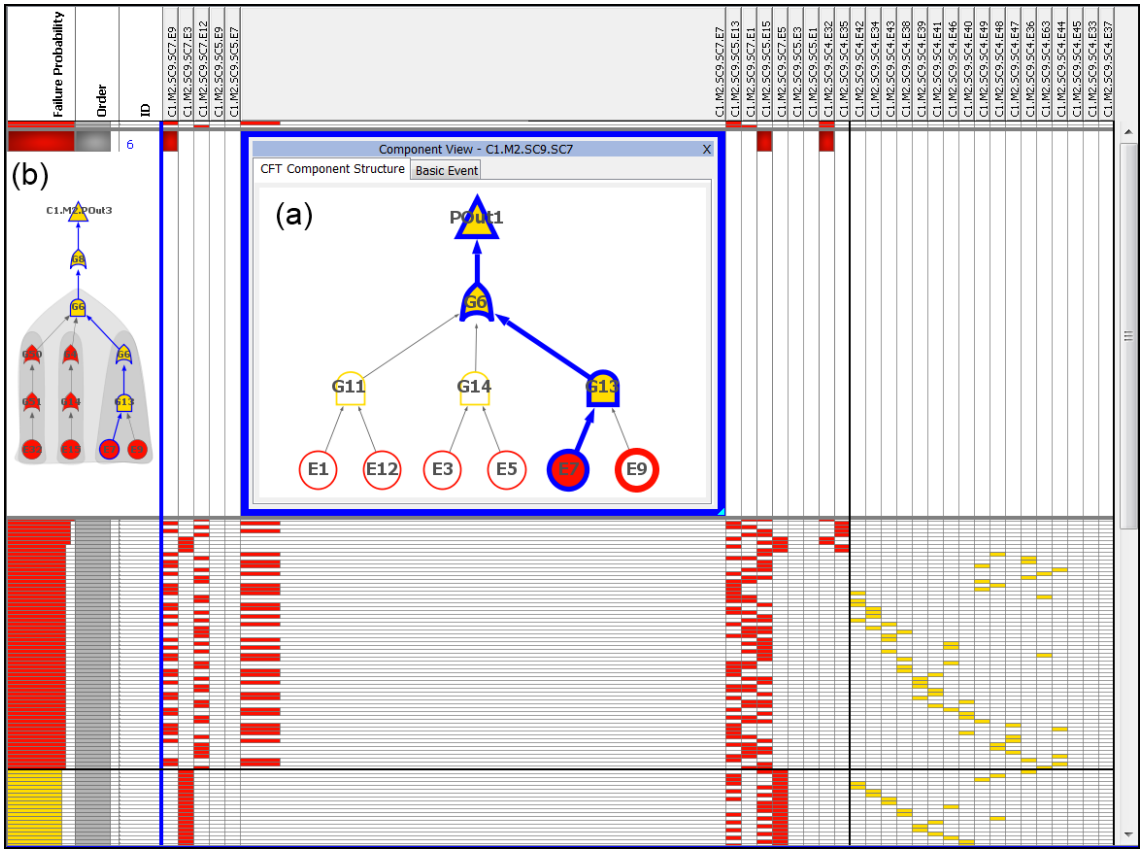


Figure 5.11: Critical path of basic events. (a) The embedded view shows the logical structure of the CFT component that contains the basic event of the current column ("E7"). The critical path is highlighted by a blue border. Nodes are assigned colors according to the unreliability levels. The nodes along the critical path are filled with colors. Other nodes only have a colored border: the basic event that is not of the current column and is included by the MCS of the current row has a thick border ("E9"); the rest of the nodes have a thin border. (b) The overview of critical paths of an MCS. The node-link layout represents the global critical paths of the current MCS. There are four basic events in the MCS. Translucent blobs represent the ranges of the CFT components. The blue border highlights the paths related to "E7". An enlarged view is presented in Figure 5.12.

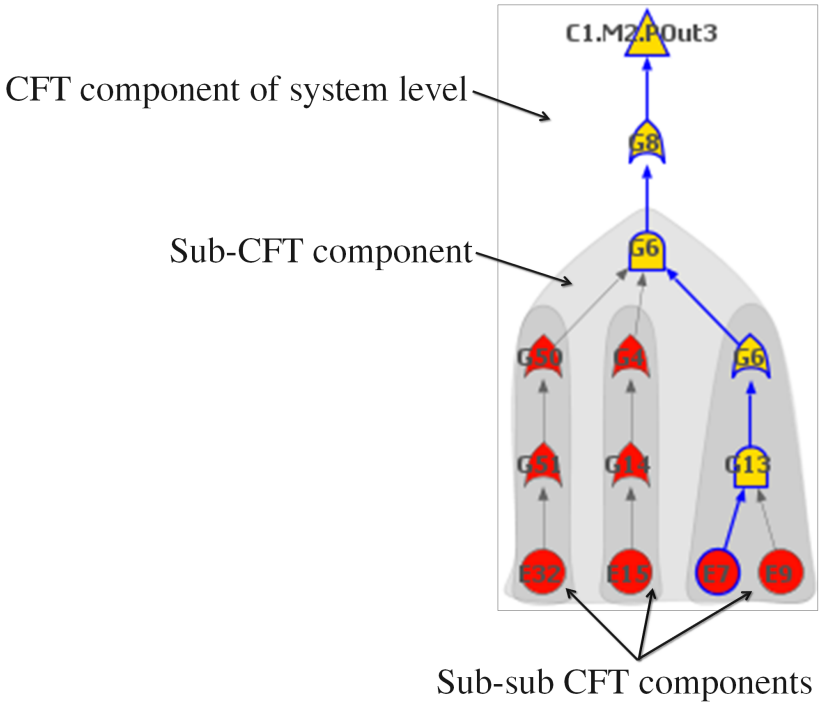


Figure 5.12: The global path view. Nesting relations among CFT components are represented in the global critical path view. The system level CFT component includes a sub-CFT component, and this sub CFT component includes three sub-sub-CFT components.

cell corresponding to the basic event will be enlarged to show the embedded view. In this way, engineers are able to analyze the logical relations between this basic event and other basic events within a CFT component while maintaining the overall global critical paths with respect to the specified MCS.

We additionally provide an alternative view for the embedded view in order to show the detail data of the currently column (Figure 5.13), e.g., parameters of the basic event corresponding to the current column. Engineers may investigate the detailed data directly in the matrix without needing to switch views.

5.1.8 Representing Relations between MCSs and CFT Components

In some cases, engineers focus on CFT components rather than basic events with respect to the following questions:

- how many and which CFT components are involved in a given MCS?
- how many and which scenarios (MCSs) may be caused by a specified CFT component?

MCS Matrix aggregates basic events according to the CFT components and merge heads of the aggregated columns as a whole to present IDs of the CFT components (Figure 5.14). Each merged column represents a CFT component. The basic events of a CFT component are represented as sub-cells in order to show the relations between

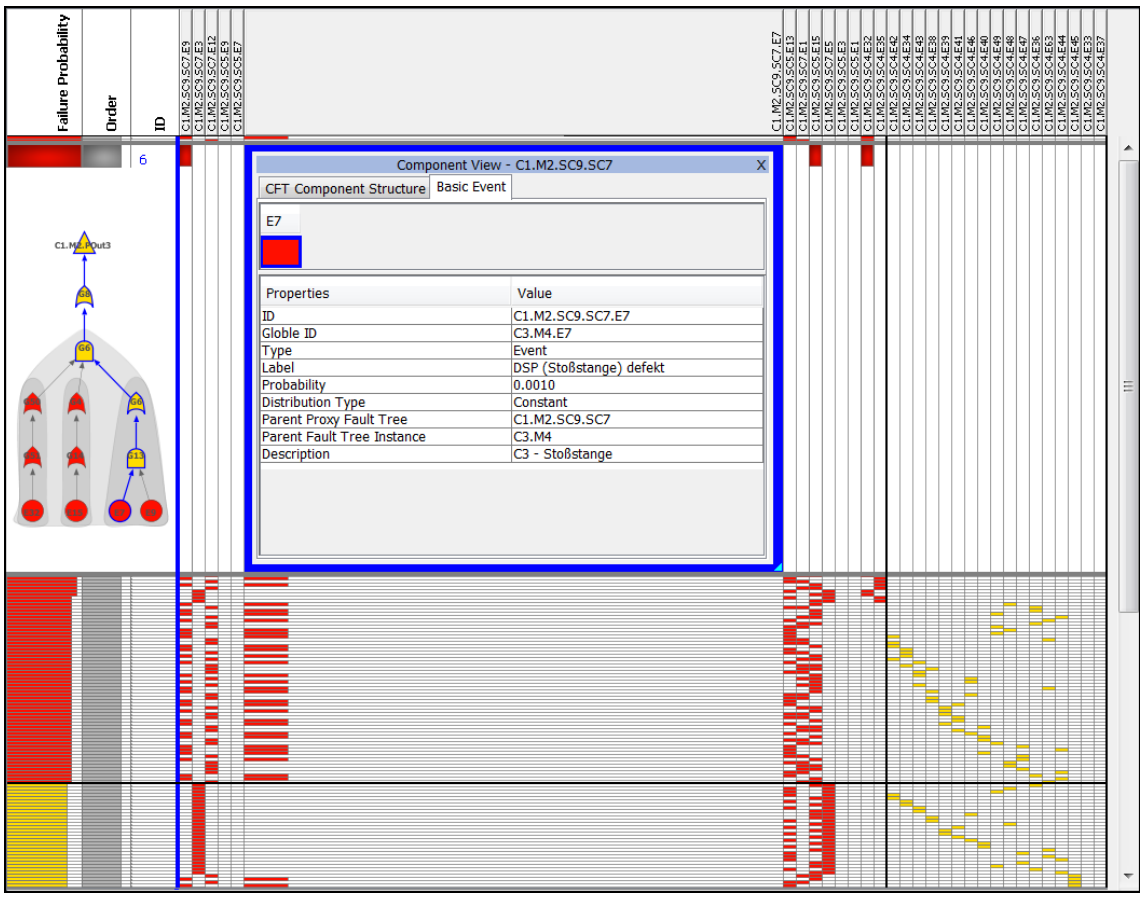


Figure 5.13: Embedded view - the detailed data of the analyzed basic event. This view consists of an icon graphically representing the basic event (upper side). The color depends on the unreliability of the basic event. A data table presents the detailed data of the basic event (lower side).

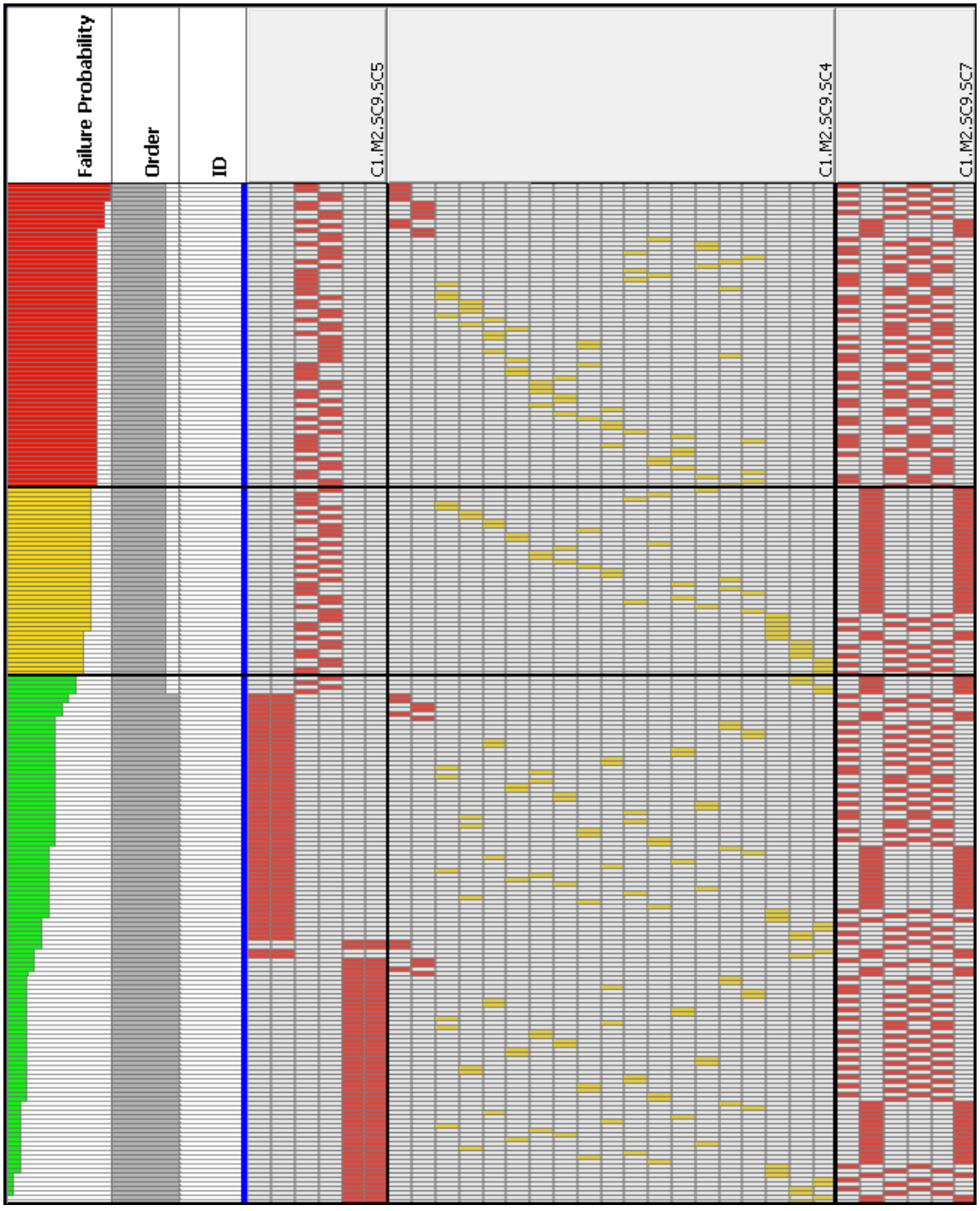


Figure 5.14: Matrix view with respect to CFT components (540 MCSs and 3 CFT components in total). Columns are aggregated and merged according to the CFT components. The merged columns represent CFT components. Basic events are represented as sub-columns.

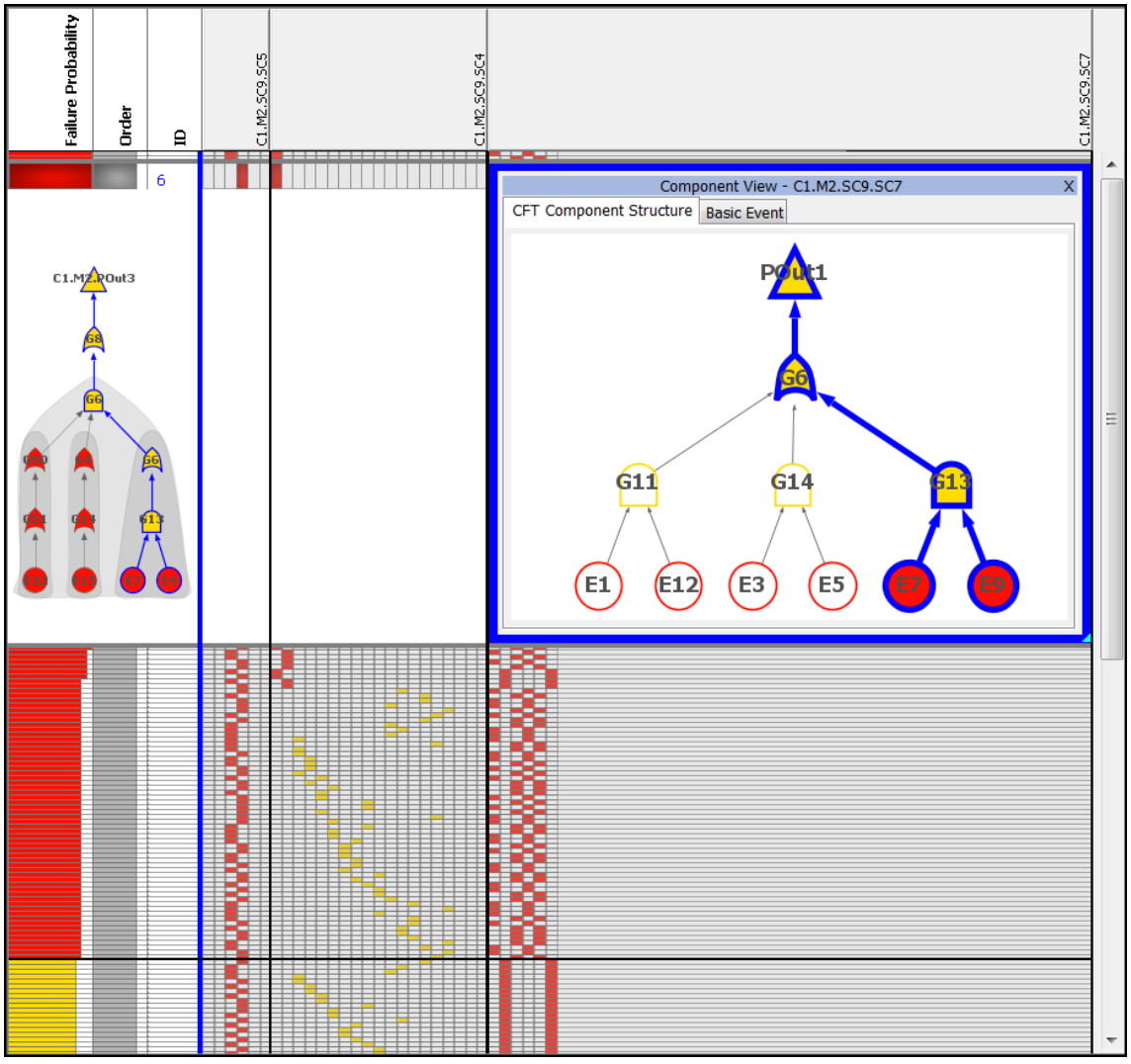


Figure 5.15: Critical paths of the CFT component. The embedded view presents the logical structure of the CFT component. The paths of the basic events "E7" and "E9" are simultaneously highlighted because both of them are included by the CFT component of the current column.

basic events and MCSs. To represent the failure propagation with respect to CFT components, the embedded views and global path view are also available (Figure 5.15). An example in section 5.2.2 depicts the analysis of the relation between MCSs and CFT components.

5.2 Application Scenarios

In this section, we present examples of the application scenarios that show the usage of our visualization approach to fulfill the tasks of MCS analysis.

5.2.1 Example 1

The overview of MCSs may show a pattern that facilitates the understanding of the system safety as well as identifying of the serious basic events. The goal of this example is to search for information from the pattern and identify the basic events that have high priority to be addressed.

5.2.1.1 Dataset and Configuration

RAVON [164] is an autonomous mobile robot made by the University of Kaiserslautern, which is used as an example by the ViERforES project [201]. A set of CFT models of RAVON was generated using ESSaRel [193] for the safety analysis. The resulting models are then explored using MCS Matrix. We apply a CFT model that was generated depending on the braking system of RAVON in the first three examples. Based on this CFT, we generate 891 MCSs and 104 basic events. The examples are presented on a monitor with a resolution of 1920x1080. The unreliability levels are defined as follows:

- critical level (red): (1e-6, 1]
- moderate level (yellow): (1e-10, 1e-6]
- acceptable level (green): (0, 1e-10]

Because the failure probabilities of the applied MCSs cover different exponential intervals, we use the logarithmic scale for the bars representing the failure probabilities. The upper bound is based on the largest failure probability, and the lower bound is the smallest failure probability. The bar representing the order of MCSs still has the default linear scale. The upper bound is the largest order of the MCSs, and the lower bound is “1”.

5.2.1.2 Analysis Process

The following steps are performed in order to accomplish the goal:

1. We group rows by the failure probability, and group columns with the associated ordering concept.
2. We apply the uniform scaling. Row height is compressed to maintain a satisfactory overview in order to analyze patterns. By analyzing the overview, as a result, Figure 5.16 shows that there are a few critical MCSs (red) and a

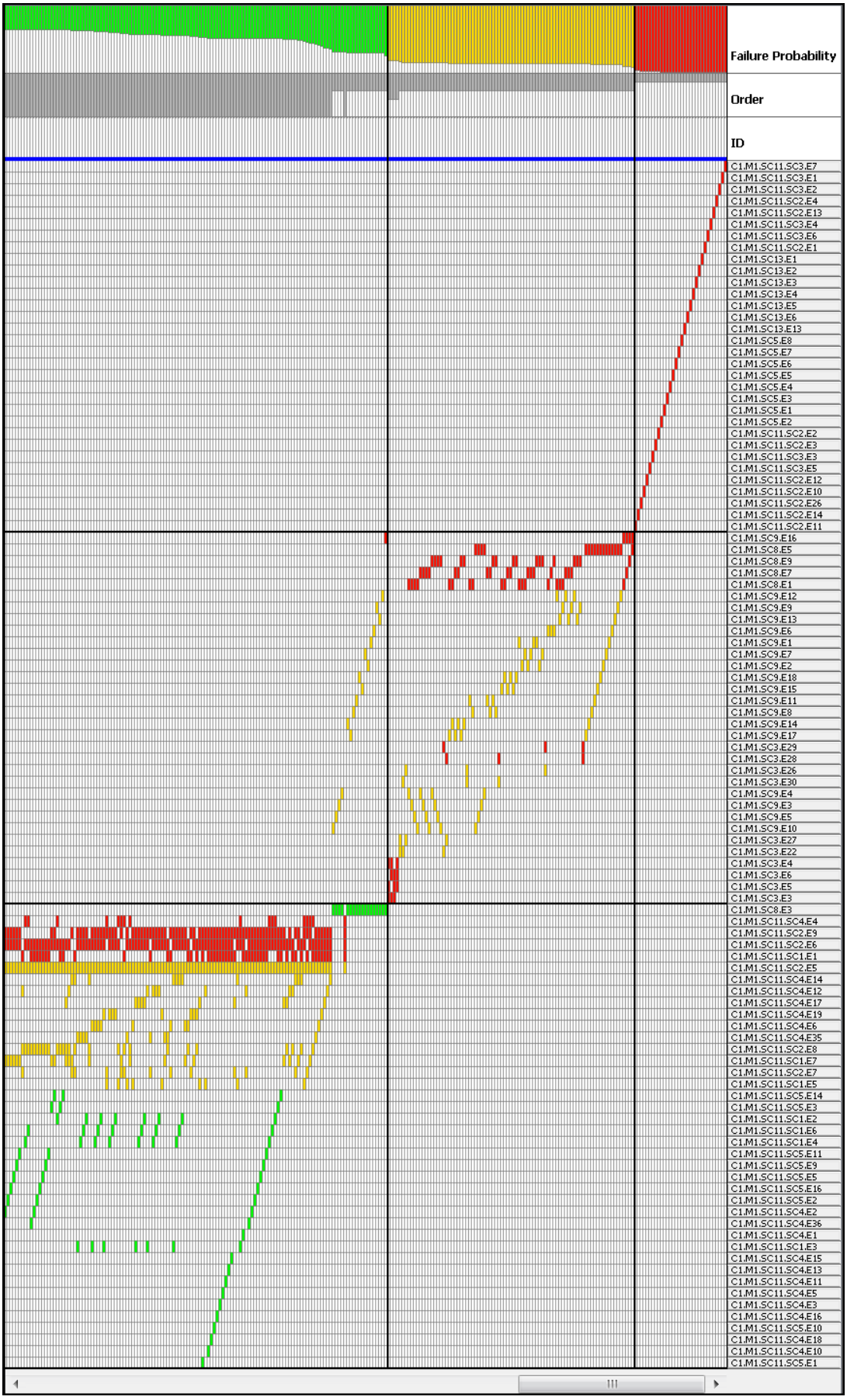


Figure 5.16: Application example 1. Overview of MCSs. (Note: only some of the acceptable MCSs are shown on the available screen space)

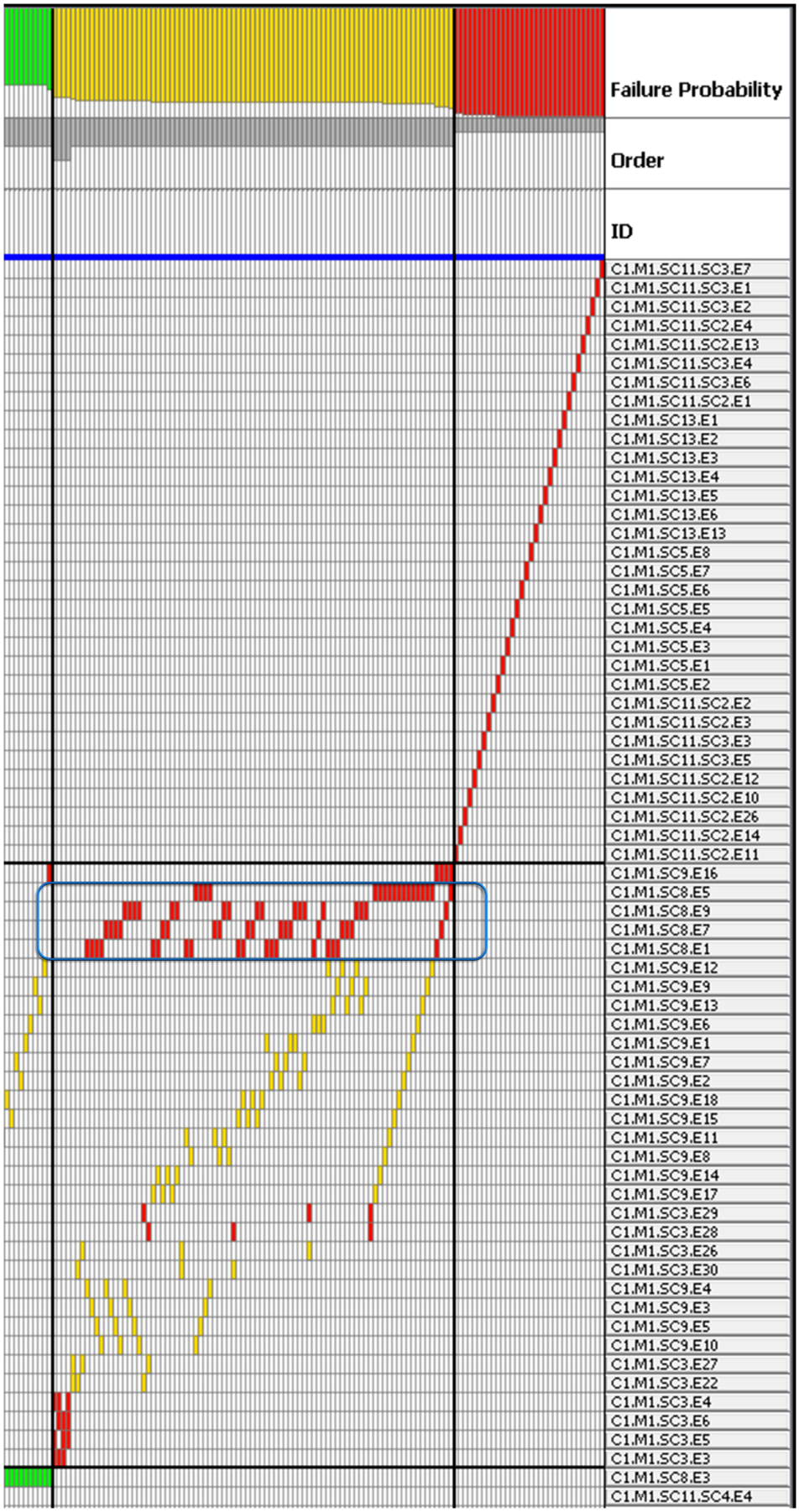


Figure 5.17: Application example 1. Critical MCSs and moderate MCSs. Four basic events frequently appear in the moderate group. The appearances are marked with the blue rectangle.

few more moderate MCSs (yellow). Most of the MCSs are acceptable (green). We may roughly understand that the safety of the overall system is not bad because most of the failure scenarios (i.e., MCSs) are acceptable.

3. Then we investigate the critical MCSs. Figure 5.17 shows the results that all the critical MCSs are the single point of failure because each MCS has only one basic event. Thus, the red rows definitely indicate the most important MCSs either by quantitative estimation (the failure probability) or by qualitative estimation (i.e., the order of MCSs). The basic events of these MCSs are dangerous and need to be addressed as soon as possible.
4. We then analyze the moderate MCSs by exploring the corresponding area. Figure 5.17 shows that there are four basic events that frequently appear in most of the moderate MCSs (manually marked with a blue frame). These basic events need to be preferentially addressed. Addressing these serious basic events may efficiently improve the MCSs of the moderate level.

By investigating the overview, the important MCSs are quickly identified. Additionally, the serious basic events are intuitively identified according to the important MCSs.

5.2.2 Example 2

In some cases, engineers analyze the relations between MCSs and CFT components. Because CFT components reflect the system components, the vulnerable system components can be identified by analyzing the CFT components. The goal of this example is to estimate the criticality of the CFT components and investigate the failure propagation within the CFT components. In this example, we focus on the critical and moderate MCSs. To achieve the goal, the following steps are needed:

1. We perform the uniform scaling for obtaining a satisfactory overview of MCSs.
2. We aggregate columns according to the inclusion relations between basic events and CFT components. In this way, columns represent CFT components (Figure 5.18).
3. We then analyze the relations between MCSs and CFT components. The following results are obtained:
   - The critical MCSs (red rows) relate to four CFT components: “C1.M1.SC13”, “C1.M1.SC11.SC2”, “C1.M1.SC11.SC3”, and “C1.M1.SC5”. The corresponding system components need to be addressed urgently.
   - The moderate MCSs (yellow rows) may be caused by three CFT components: “C1.M1.SC9”, “C1.M1.SC8”, and “C1.M1.SC3”. Specifically, most of the moderate MCSs are caused by the combinations between CFT components “C1.M1.SC9” and “C1.M1.SC8”. As long as one of these components is addressed, the system safety may be improved.
   - The CFT components “C1.M1.SC11.SC4”, “C1.M1.SC11.SC5”, and “C1.M1.SC11.SC1” are only related to the acceptable MCSs. The failures of the corresponding system components do not seriously influence the system safety.

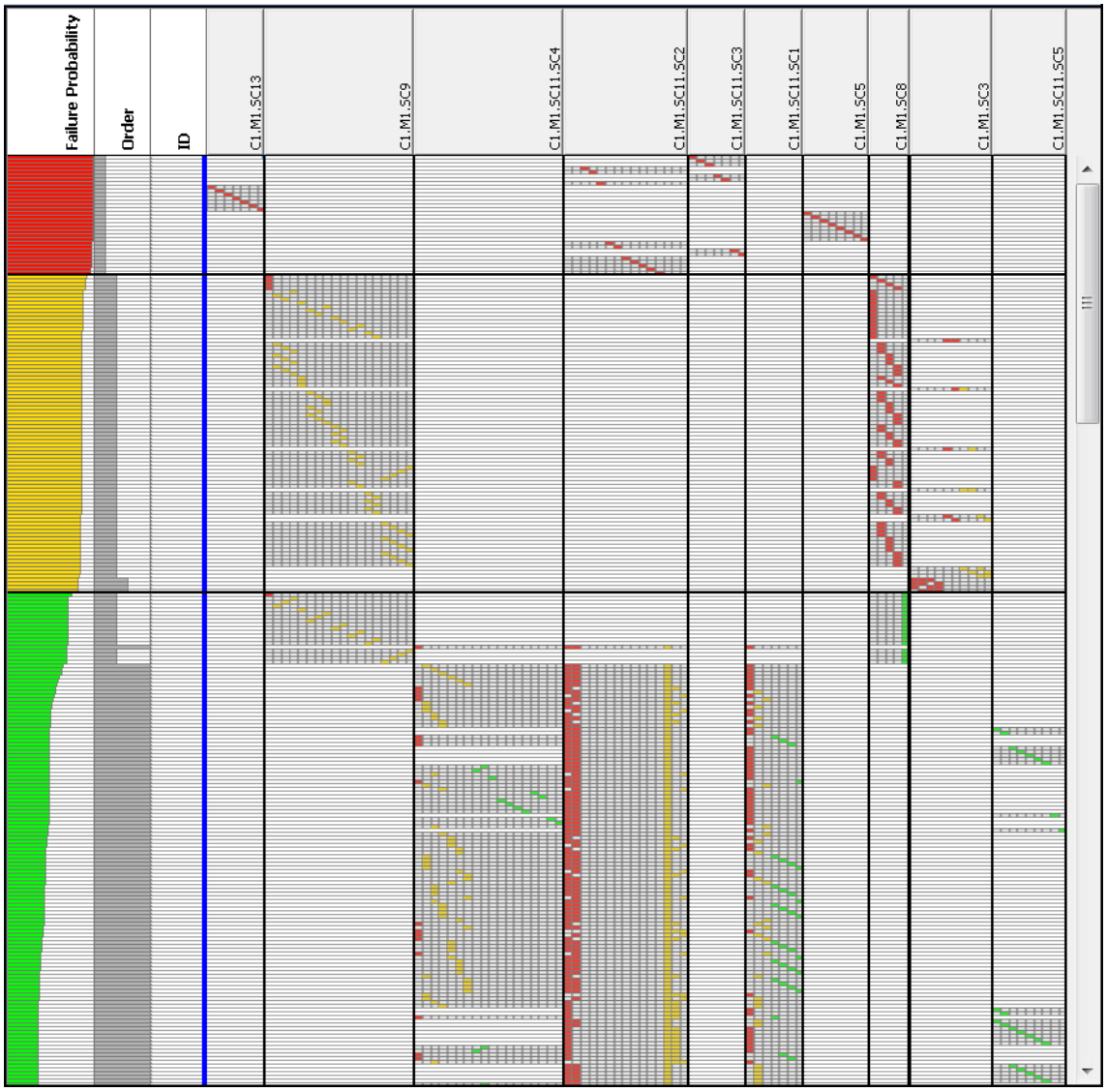


Figure 5.18: Application example 2. Columns represent the CFT components.

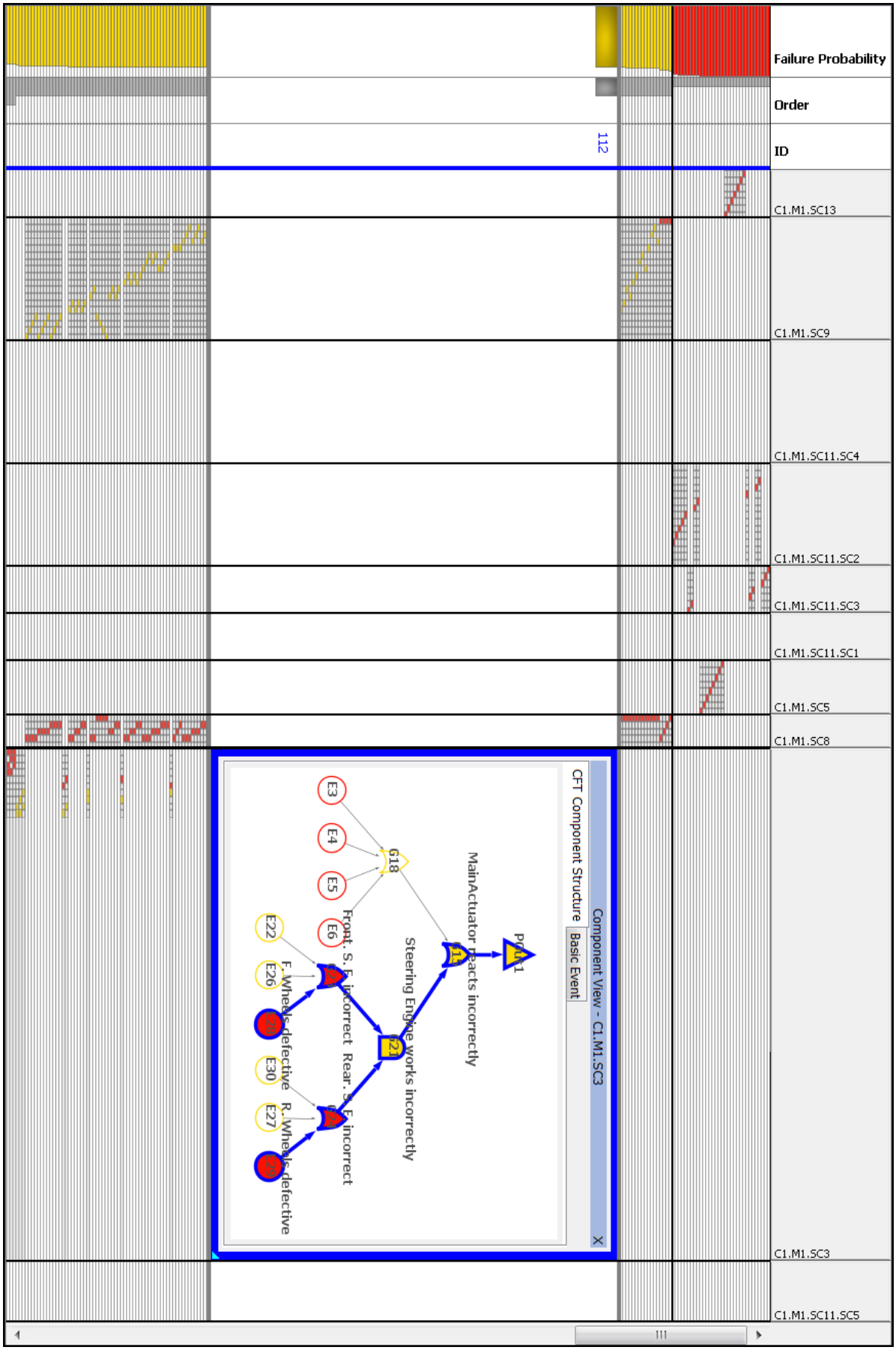


Figure 5.19: Application example 2. Internal structure of the CFT component.

4. We finally investigate the internal failure propagation of the specific CFT component. As an example, we focus on the CFT component "C1.M1.SC3" that may cause MCSs without cooperating with other components. This component represents the main actuator of the breaking system of the robot. We show the structure of this CFT component in the embedded view (Figure 5.19). By dynamically displaying the labels of nodes, the meaning of the failure flow inside this component may be understood as follows. There are two basic events that respectively represent the problems of "Front wheels (E28)" and "Rear wheels (E29)". Both of the basic events cause the intermediated failures of the front steering engine and the rear steering engine. When both steering engines fail at the same time (i.e., at gate "G21"), the main actuator cannot correctly provide the reaction (i.e., at gate "G15").

The example illustrates the pattern analysis for relations between MCSs and CFT components. The CFT components are treated as the entities contributing to MCSs. This helps to identify the vulnerable system components that may cause the critical failures of a system.

5.2.3 Example 3

For an MCS, failure propagation of its basic events describes how the top event is triggered in the failure scenario of this MCSs. The failure propagation shows the generated intermediate failures and the failed CFT components in the scenario. Because the CFT component reflects the system components, the failure propagation analysis may facilitate the analysis of the impacts on the system components in case a specific failure scenario appears. This example is presented for this application. The goals are: identification of the most critical MCS(s) that have the largest failure probability; identification of the failure propagation cased by the MCS(s). The following steps are performed in order to fulfill these goals:

1. We investigate the red rows in order to identify the most critical MCS. However, the failure probabilities of the first three MCSs look very similar.
2. We then perform the individual scaling and investigate the failure probabilities of the first three MCSs (Figure 5.20). This way, texts clearly show that only the first MCS, which has the ID "15", has the largest failure probability. The basic event included by this MCS is "C1.M1.SC11.SC3.E7".
3. By double-clicking the color-filled cell at the intersection between the MCS "15" and its basic event, embedded view is displayed that shows the logical structure of the parent CFT component of the basic event (Figure 5.21).
4. The critical path of the basic event included by the MCS "15" is highlighted with a blue border in the embedded view: "E7 → G11 → G8". The meaning of the failure propagation is identified by node labels: "Real-time detection issue (E7)" → "Rotation Direction issue (G11)" → "Unfiltered 3D Obstacle Detection (G8)".
5. We then analyze the failure propagation between CFT components in the global path view (Figure 5.22). The top event of the system-level component may be caused by the following failure propagation:

Figure 5.20: Application example 3. The identifying the most critical MCS. Using the individual scaling to confirm the MCS having the largest failure probability. The MCS "15" is the unique resulting MCS. It includes the basic event "C1.M1.SC11.SC3.E7" (marked with a circle).

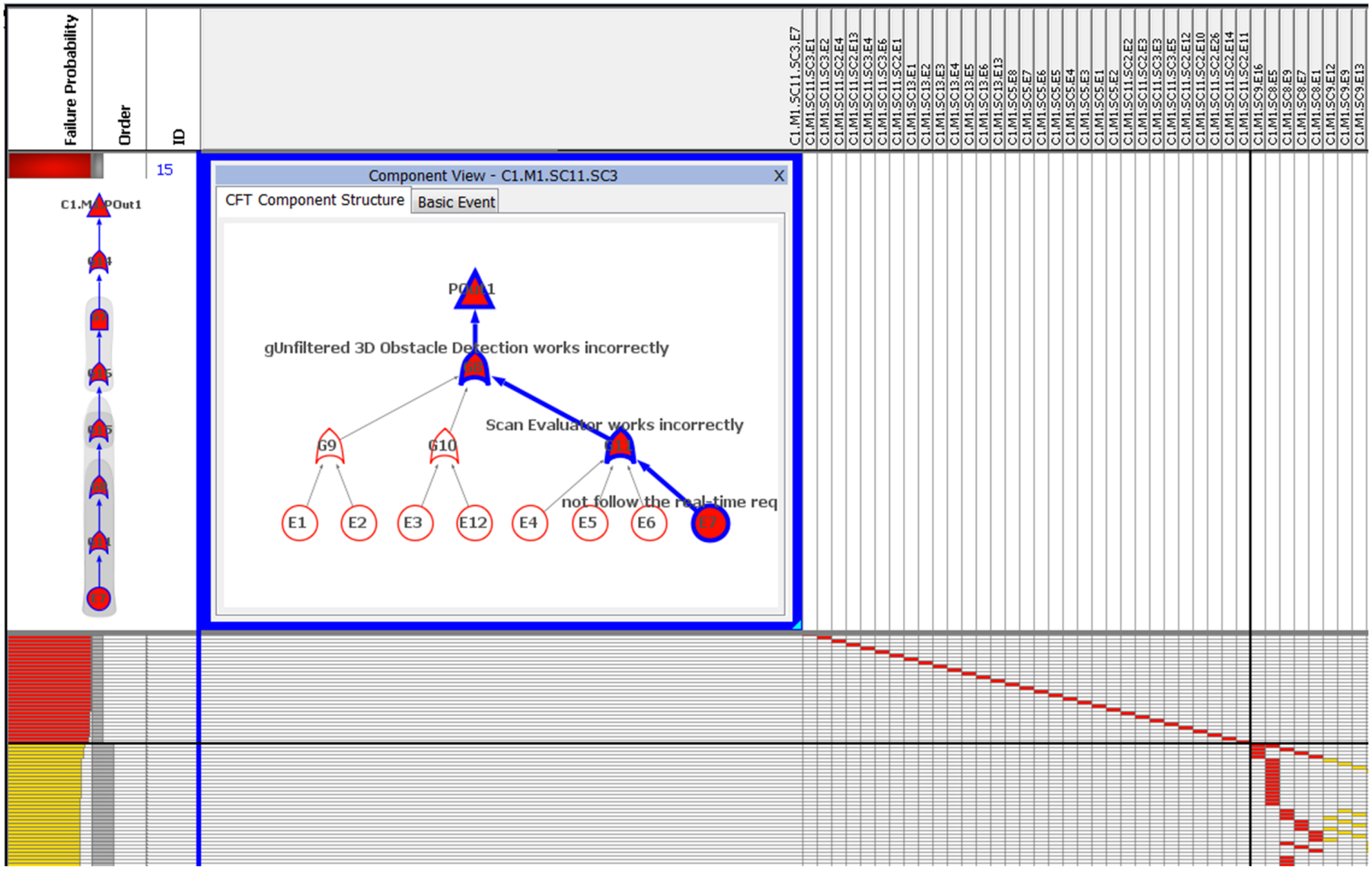


Figure 5.21: Application example 3. The identification of the logical structure of the relevant CFT component. The logical structure of the CFT component contains the identified the basic event "C1.M1.SC11.SC3.E7". The critical path of the basic event is highlighted using a blue border.

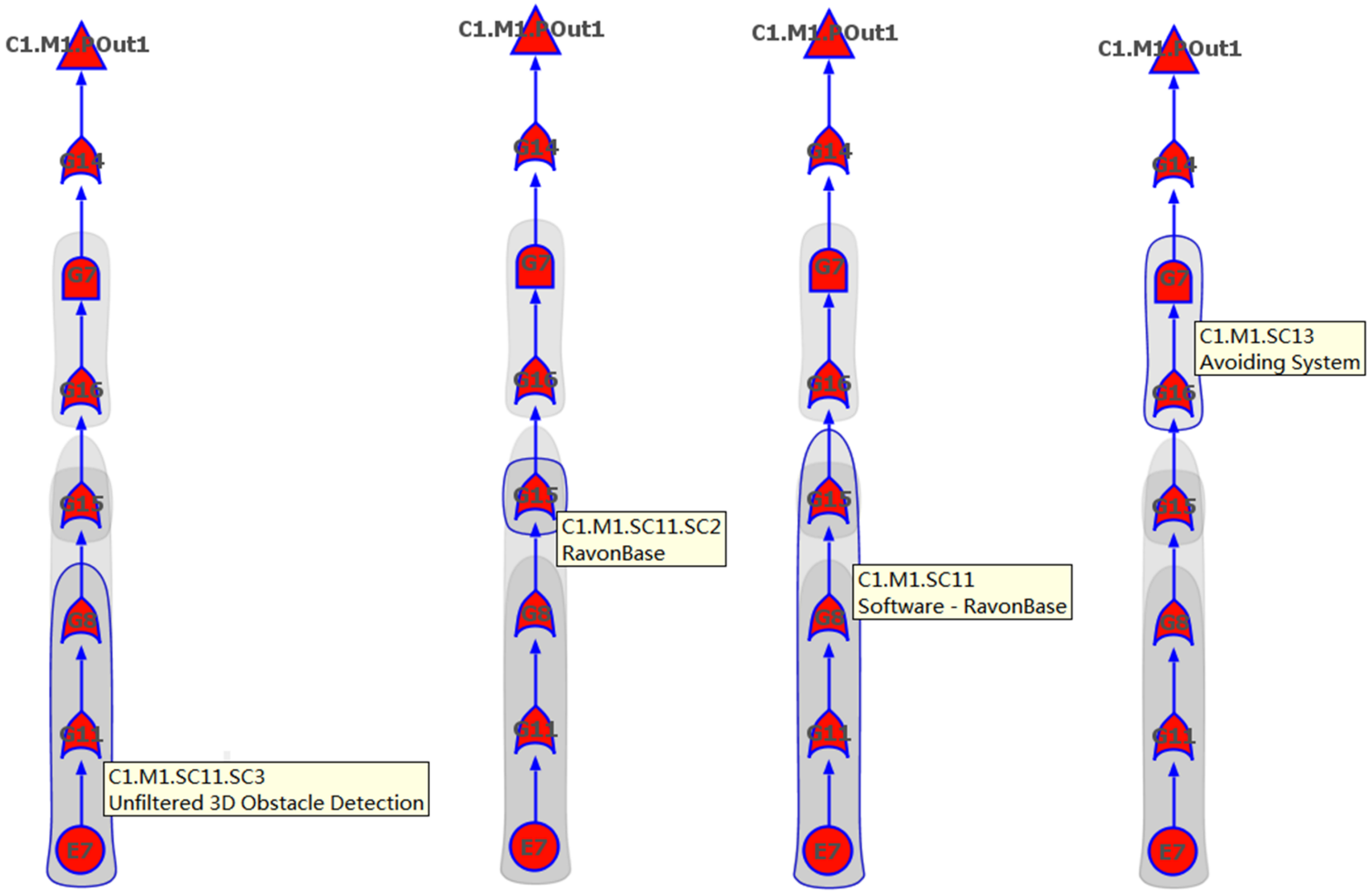


Figure 5.22: Application example 3. Nesting relations and failure propagation between CFT components. The selected CFT component is indicated by a blue border of the blob. (a) The CFT component “Unfiltered 3D Obstacle Detection”. (b) The CFT component “RavonBase”. (c) The CFT component “Software - RavonBase”. (d) The CFT component “Avoiding System”.

- The CFT component "C1.M1.SC11.SC3" outputs its top-level failure to CFT component "C1.M1.SC11.SC2". An intermediate failure is generated at gate "G15" (Figure 5.22 (a) and (b)).
- The CFT components "C1.M1.SC1.SC3" and "C1.M1.SC11.SC2" are contained by CFT component "C1.M1.SC11" (Figure 5.22 (c)).
- The CFT component "C1.M1.SC11" outputs its failure to the CFT component "C1.M1.SC13" (Figure 5.22(d)). Inside this CFT component, the intermediate failure causes the output failure by gate "G7".
- The failure from the CFT component "C1.M1.SC13" leads to an intermediate failure at gate "G14" of the system-level CFT component. The failure at gate "G14" directly leads to the top event.

In short, the paths with respect to the meaning of the failure propagation is described as follows: the failure of the system component "Unfiltered 3D Obstacle Detection" flows into the part of the robot "RavonBase", and causes the software failure of this part. Then the failure causes the problem of the avoiding system of the robot. Finally, the mobile robot may not be able to break because of the failure of the avoiding system.

This example shows how to analyze the failure propagation of MCSs using the node-link critical paths. Additionally, the example again presents the use of the individual scaling in terms of the exact comparison in the analysis.

5.2.4 Example 4

The MCS analysis with respect to a specific basic event is significant in many scenarios. A significant case is to deal with a specific basic event that has a high failure probability and is difficult to improve. In this case, engineers need to intensively address the partner basic events that may cooperate with the specific basic event to cause failure scenarios (i.e., MCSs). This example illustrates this scenario. We estimate the basic event according to the failure probability. The objective of this example is to identify the partner basic events of the most critical basic event(s).

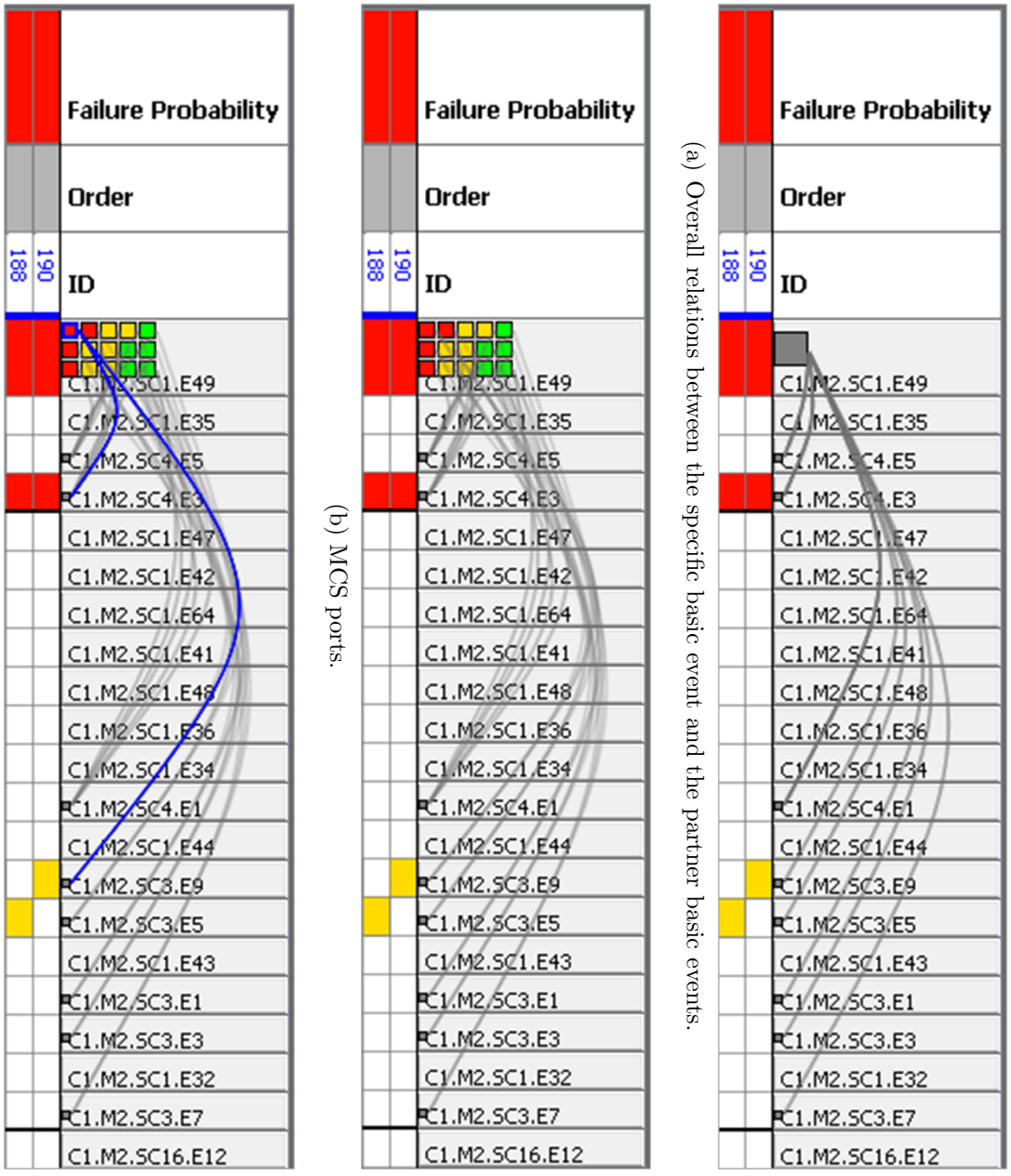
5.2.4.1 Dataset and Configuration

We use a CFT model based on RAVON that contains 52 basic events. There are 212 MCSs generated based on this CFT. The configuration is the same as that of the data used in the first three examples.

5.2.4.2 Analysis Process

To achieve this goal, the following steps are required:

1. We group columns by the failure probability in order to rank the basic events. As a result, the basic event "C1.M2.SC1.E49" of the first column is the most critical (Figure 5.23 (a)).
2. We show the relational graph of the most critical basic event (Figure 5.23 (a)). As a result, the partner basic events are identified by the partner-BE ports along the curved lines on the matrix header:



(a) Overall relations between the specific basic event and the partner basic events.

(b) MCS ports.

(c) The most significant partner basic events.

Figure 5.23: Application example 4. The relations between the basic events.

- the basic event "C1.M2.SC4.E5"
- the basic event "C1.M2.SC4.E3"
- the basic event "C1.M2.SC4.E1"
- the basic event "C1.M2.SC3.E9"
- the basic event "C1.M2.SC3.E5"
- the basic event "C1.M2.SC3.E1"
- the basic event "C1.M2.SC3.E3"
- the basic event "C1.M2.SC3.E7"

3. We show and analyze the MCS ports of the most critical basic event. Figure 5.23 (b) shows the following results. There are four critical MCSs (red ports) including the critical basic event. There are six moderate MCSs (yellow ports). We do not need to focus on the partner basic events connecting the green ports because the risk of the relevant MCSs are acceptable.
4. We then investigate the most significant partner basic events (Figure 5.23 (c)). Because the MCS ports are sorted by the failure probability, the first red port (at the lower left corner) represents the most critical MCS. By selecting the port, curved lines indicate the partner basic events included by the most critical MCS: "C1.M2.SC4.E3" and "C1.M2.SC3.E9". These are the most significant basic events that may cooperate with the most critical basic event to cause the most critical failure scenario.

The example presents the identification of the partner basic events of a specific basic event by using the relation ports. Engineers may identify the relations between basic events without needing to investigate the cells of the matrix view.

5.3 Evaluation

In order to evaluate our visualization approach, we performed a user experiment. The experiment focuses on the completion time for efficiency analysis and accuracy of MCS analysis for effectiveness analysis. The terms *efficiency* and *effectiveness* are defined in [98]:

- efficiency is defined as the relationship between the result achieved and the resources used.
- effectiveness is the extent to which planned activities are realized and planned results achieved.

5.3.1 Hypotheses

Hypotheses for the experiment were defined as follows.

- Hypothesis 1: MCS Matrix is more efficient than the reference tool. It relates to the efficiency by measuring the completion time. The null hypothesis $H_{1,0}$: MCS Matrix is NOT more efficient than the reference tool.
- Hypothesis 2: MCS Matrix is more effective than the reference tool. It focuses on the effectiveness by measuring the accuracy. The null hypothesis $H_{2,0}$: MCS Matrix is NOT more effective than the reference tool.

5.3.2 Experiment Environment

5.3.2.1 Reference tool.

ESSaRel [193] was used as the reference tool because it perfectly supports the CFT analysis and the MCS analysis.

5.3.2.2 Participants.

There were 14 participants from the local university who have Master's degrees in computer science or engineering. They did not have any experience with the tools prior to the experiment. The participants were divided into two groups equally: group A using MCS Matrix and group B using ESSaRel.

5.3.2.3 Apparatus and software.

The experiment was performed on a computer with an AMD Athlon 64 x2 Processor 5000+ with 2GB of memory and an NVIDIA Quadro NVS210S graphic card with 64MB video memory. The tools were running on a 19" TFT monitor with a resolution of 1680x1050. Windows XP and the necessary software were installed on the computer. The applied tools were initially configured and would not change the settings during the experiment, so that all participants can have the same initial software environment.

5.3.2.4 Dataset.

We applied a CFT model of the robot RAVON. The MCSs were generated using ESSaRel and then visualized using MCS Matrix. Under consideration of the workload and the actual difficulties of the experiment, e.g., the duration of each participant, a CFT model having 118 MCSs for the experiment were selected and applied.

5.3.3 Experiment Procedure

Initially, an inner test experiment was performed in order to check whether the experiment design was feasible. After correcting the found problems, two domain experts were invited for an expert review. They provided feedback not only about our tool, but also about the experiment design, such as the difficulty of the task and the time needed for completing the task. After the modifications made based on the experts' comments, the participants carried out the evaluations one at a time.

There was a moderator and an observer present. The moderator focused on the process of evaluation. He guided the participants through all steps of the evaluation and answered their questions. The observer was responsible for recording the completion time of each step and the behavior of subjects in protocols. The tool was restarted at the beginning of every experiment in order to restore the software to its initial state.

5.3.3.1 Training.

At the beginning of the experiment, the participants were asked to read tutorials for the tools. Then we allowed them to experience the tools for a trial period. The time of the training phase was recorded but not limited in order to give participants enough time to become familiar with the tools.

5.3.3.2 Task.

After training, participants started to perform a task. Both groups fulfilled the same task. The goal of the task was to draw the logical structures of the CFT components that relate to the MCSs having the critical failure probability. The participants completed the task by accomplishing the following sub-tasks:

- Identify the MCSs that have a critical unreliability level, i.e., the critical MCSs.
- List the failure probabilities, the orders, and the basic events of the critical MCSs.
- Identify the most critical ones from the identified basic events.
- Draw logical structures of the CFT components that have the most critical basic events.

In the task, the tools may be analyzed in two essential aspects of the MCS analysis:

- aspect 1: identification of the important information of MCSs (the sub-task 1, 2, 3).
- aspect 2: investigation of the failure propagation of MCSs along the CFT structure (sub-task 4).

The time of all sub-tasks was recorded but not limited in this phase. The participants stopped working when they were completely satisfied with their results.

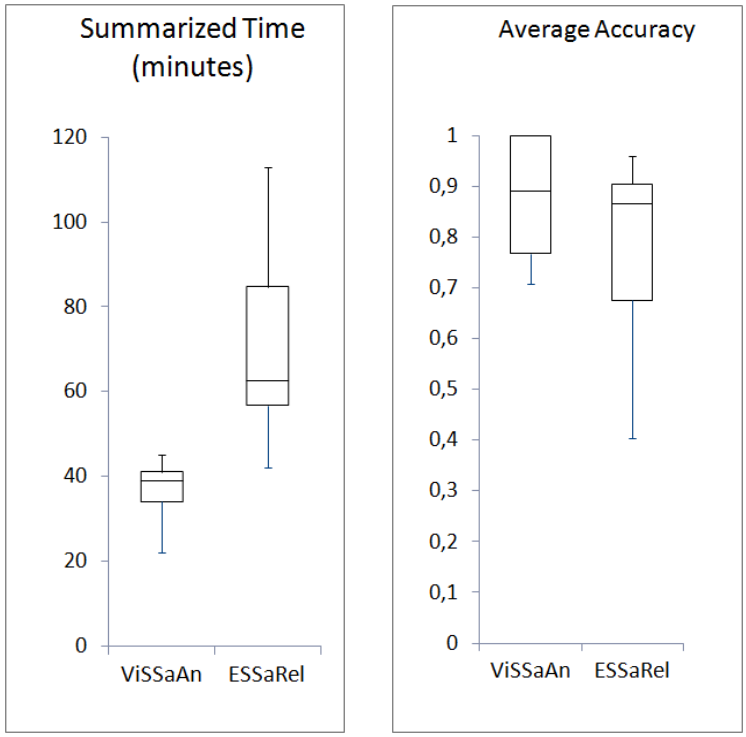
5.3.3.3 Questionnaires.

Finally, participants were asked to fill out a questionnaire for the qualitative evaluation. It had a five-point Likert scale to represent how good/bad the achievements of the tools were.

5.3.4 Results and Discussion

5.3.4.1 Quantitative Evaluation

The efficiency was evaluated by analyzing the average summarized time for completing the task. Group A using MCS Matrix spent 37 minutes of average on the task, and group B using ESSaRel spent 72 minutes. Using MCS Matrix, the task was completed 48.61% faster than using ESSaRel (Figure 5.24 (a)). In order to identify the significant difference between both results, an ANOVA F-test [129] was performed. The result of the F-test: $F = 9.655$, $p = 0.011$. Because $p < 0.05$, the difference in the time needed for completing the task was significant. The null hypothesis $H_{1,0}$ with respect to the efficiency was rejected. We deduced that MCS Matrix was more



(a) Summarized time. (b) Average accuracy.

Figure 5.24: Quantitative results of the experiment.

efficient than ESSaRel. The time of the sub-tasks was also summarized in order to analyze the tools in different aspects. For aspect 1, using MCS Matrix, the sub-tasks were completed 47.78% faster than using ESSaRel. By ANOVA F-test, the results were $F = 7.294$, $p = 0.022 < 0.05$. We deducted that it was significantly faster using MCS Matrix than using ESSaRel for identifying the important MCSs and the associated information. For aspect 2, using MCS Matrix, the sub-task was completed 62.5% faster than using ESSaRel. By ANOVA F-test, the results were $F = 6.585$, $p = 0.028 < 0.05$, and thus we deducted that it was significantly faster for identifying the failure propagation of the MCSs using MCS Matrix than using ESSaRel.

Further, effectiveness was evaluated by analyzing the average accuracy of the task answers. Group A using MCS Matrix had 87.5% of the average accuracy, and group B using ESSaRel had 77% of the average accuracy. That is to say, the MCS Matrix group had a 10.5 % higher accuracy than that of the ESSaRel group (Figure 5.24 (b)). The results of an ANOVA F-test were $F = 1.012$ and $p = 0.338$. Because of $p > 0.05$, there was no significant difference in accuracy between both results. The null hypothesis $H_{2,0}$ with respect to the effectiveness had to be accepted. We deduced that the accuracy of MCS Matrix was not significantly better than ESSaRel. Analyzing the time in individual aspects, there was no significant difference.

In short, we concluded that when using MCS Matrix, engineers may analyze MCSs much faster than using traditional approaches and obtain the results with similar accuracies.

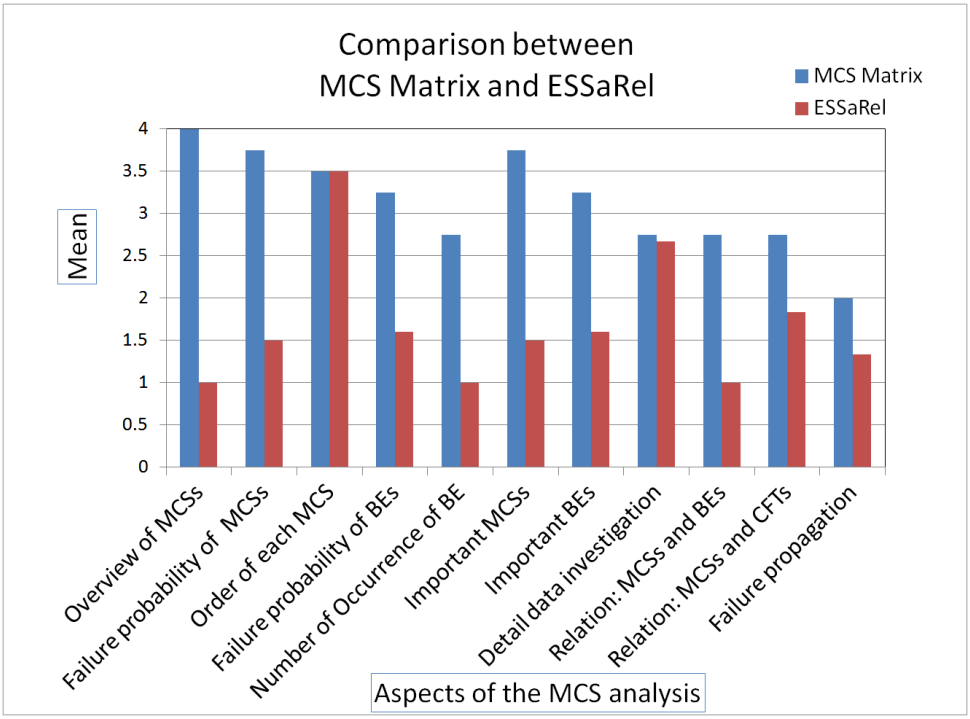


Figure 5.25: Subjective comparison between MCS Matrix and ESSaRel.

5.3.4.2 Qualitative Evaluation

A Likert-scale questionnaire was designed for the comparison between MCS Matrix and ESSaRel. The items of the questionnaire depended on the tasks of the user experiment. The participants subjectively evaluated the tools according to the abilities of tools for solving the problems. We assign points from 0 to 4 to the estimation levels from "strongly disagree" to "strongly agree". The better the comment, the higher the points awarded. The mean of the points were calculated and shown in Figure 5.25. It showed that most items was favorably reviewed.

The participants basically commented that MCS Matrix was helpful to obtain a satisfactory overview of MCSs that was helpful for analyzing the pattern of MCSs, e.g., roughly estimating the current risk state of a system. Additionally, they would like to navigate MCSs in the matrix view rather than in plain text. The participants commented that colors and the grouping concepts can effectively support to estimate MCSs (and basic events), and identify the important information. The concept combining the uniform scaling and the individual scaling was favorably reviewed. The participants thought that it was practical to view the detailed data of both MCSs and basic events in the overview of a large number of MCSs. The "scaling by groups" was preferred but not very greatly. The participants argued that the benefit of the concept would be reduced when there were a lot of critical MCSs. In this case, the screen space would be occupied by the rows of the critical MCSs, so the overview of the MCSs would be damaged. The participants thought that the individual scaling is more practical than the scaling by groups. The participants considered that they can intuitively and quickly compare the MCSs using bars. Most of the participants commented that representing the critical path inside the matrix view facilitates the

analysis of failure propagation because this concept reduces the effort for switching different views. Some participants argued that the embedded view takes up too much space and reduces the readability of the overview of MCSs. They suggested a pop-up window for representing the requested logical structure.

In addition, the participants thought that MCS Matrix was able to effectively represent the relations among basic events with respect to the common MCSs. Representing the relation between MCSs and CFT components using a matrix was treated as a novel concept because there were few considerations of this relation in the ordinary concepts. Generally, the participants commented that MCS Matrix had great benefits in terms of the identification of the important information from a large number of MCSs.

Chapter 6

Visualization of Importance Analysis

The importance analysis is a significant quantitative measure that evaluates the respective contributions of basic events to the top event. According to the design concepts discussed in section 4.2, we propose a visualization approach, called VisQSA (Visual Quantitative Safety Analysis), which visually supports the importance analysis.

6.1 Visual Quantitative Safety Analysis

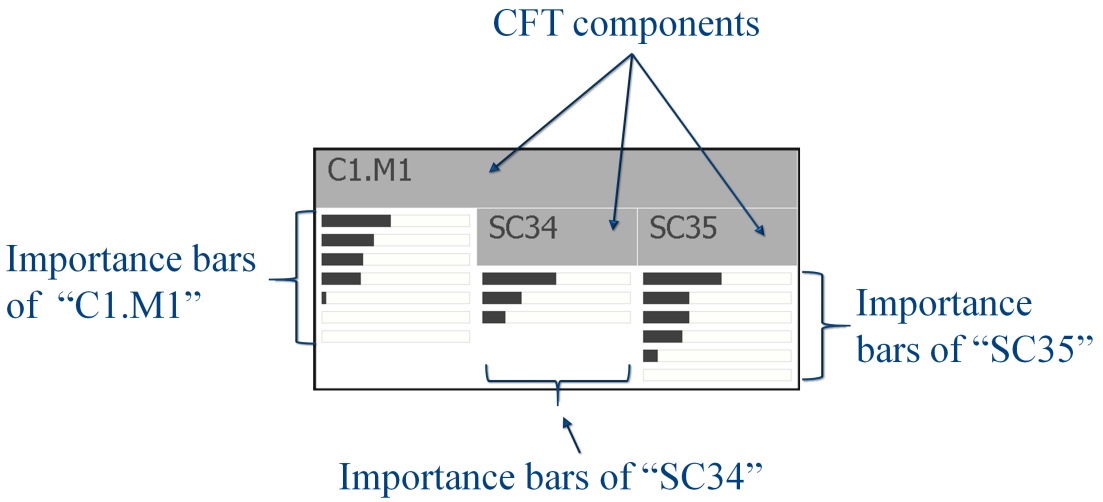
Our approach dynamically integrates the architectural view representing the results of the importance analysis with the enhanced node-link logical structure of CFT components representing logical failure flow.

6.1.1 Architectural View

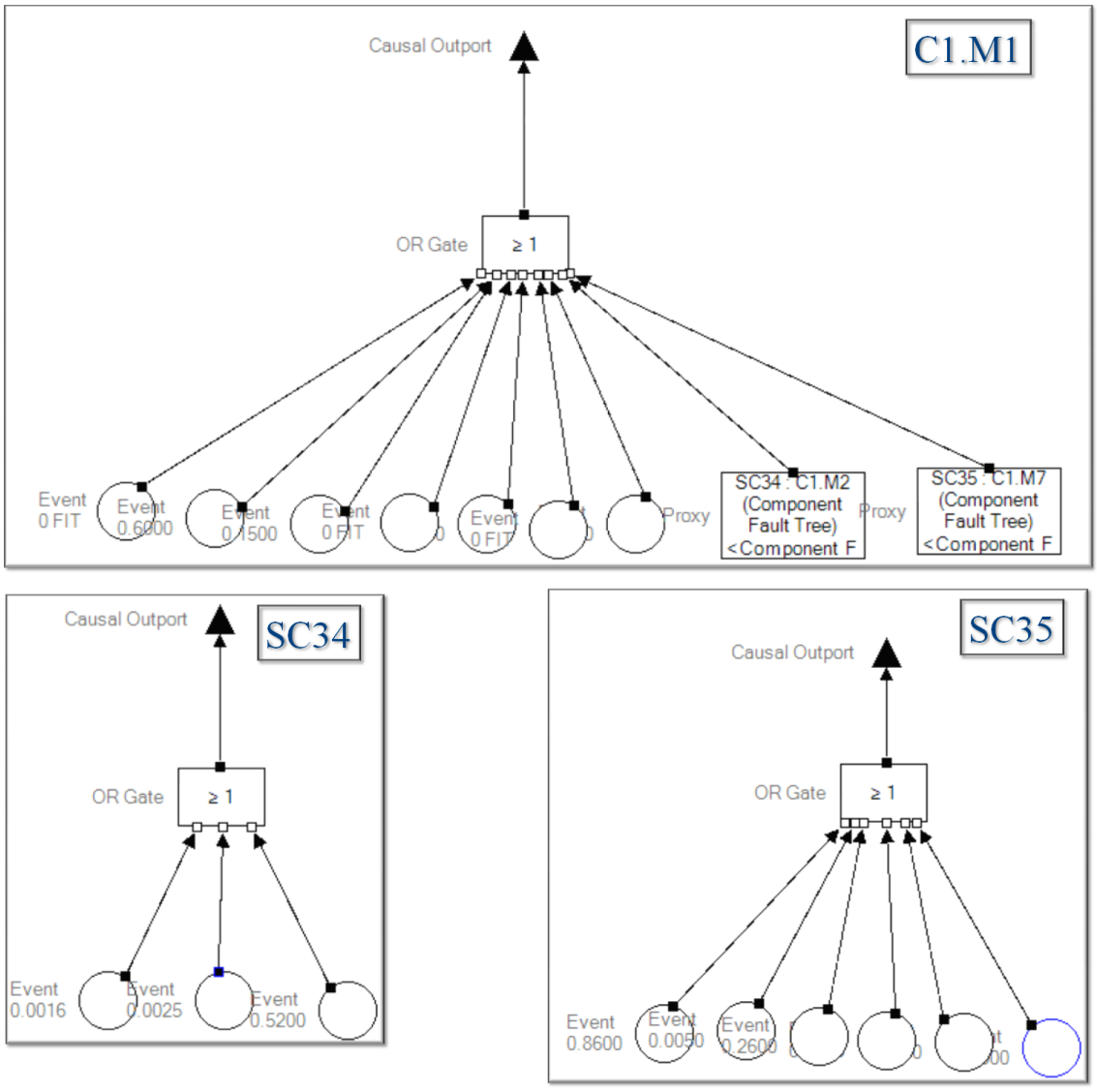
We propose an iceray-layout-style architectural view (Figure 6.1) for representing the nesting relation between CFT components. A gray rectangle represents a CFT component. The ID of the CFT component is printed on the left side of the rectangle. Basic events with importance values are represented using the importance bars. The filled part of a bar represents the importance value of the basic event. The importance bars are sorted in descending order and vertically listed under the parent rectangle on the far-left side. The rectangles representing the sub-components are horizontally listed beside the list of importance bars.

In some cases, the importance values for a system are too small to be represented. To address this issue, engineers are allowed dynamically to define the lower and upper bounds of the importance bar between 0 and 1. An alternative logarithmic scale may be applied to the importance bar when the importance of basic events are distributed in a large exponential interval, e.g., [0.001, 0.1]. The architectural view represents various situations of the containing relations (e.g., Figure 6.2):

- the CFT component contains both basic events and sub-CFT components, e.g., the CFT component “SC43”.



(a) Architectural view. Gray rectangles represent CFT components. The basic events with importance values are represented using the importance bars.



(b) Ordinary representation concept (generated using ESSaRel [193]). The logical structures of CFT components are presented in three separate views.

Figure 6.1: Architectural view. (a) The architectural view. (b) The ordinary representation of the CFT model represented in (a).

Figure 6.2: Different cases of the containing relations.

- the CFT component does not contain basic events, but has sub-CFT components, e.g., the system-level CFT component "C1.M1".
- the CFT component does not contain sub-CFT component, but has basic events, e.g. "SC34", "SC35", "SC105", and "SC109".
- the CFT component has neither basic events nor sub-CFT components, e.g., "SC28".

6.1.2 Integrating the CFT Structure

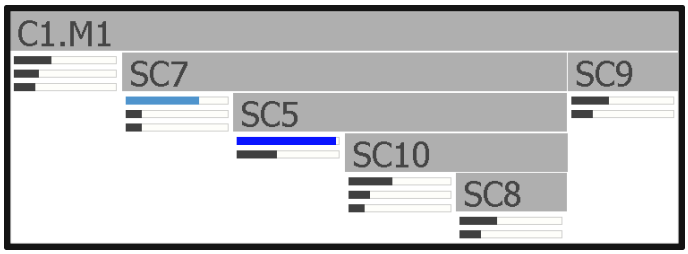
The logical structure of the CFT component is the basis of representing the failure flow and logical relations between basic events. According to the layout composition rules (section 4.2.3.3), the logical structure of the specific CFT component is displayed, when requested. The visible logical structure may be treated as the expansion of the architectural view. In order to adapt to different demands, we provide two expansion concepts: referencing expansion and in-place expansion.

6.1.2.1 Referencing expansion.

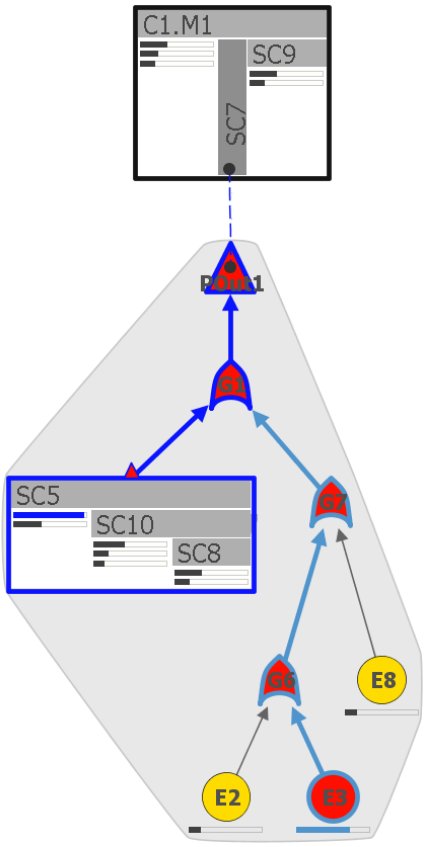
The referencing expansion concept (Figure 6.3 (b)) is developed based on case 2 (d) in Table 4.7. When expanding a specific CFT component, the importance bars and the architectural views of the sub-components are embedded in the visible logical structure. In the architectural view, a vertical rectangular indicator is drawn to replace the initial part representing the CFT component. A dashed curve links the visible logical structure and the indicator. By using the ordinary representation concept, when investigating a desired sub-CFT components, engineers need to show all the higher-level parent components as the example shown in section 3.2.2.3. If this component is nested in a deeper hierarchy, it may take a lot of effort for the iterative investigation. In contrast, using the referencing expansion, engineers are allowed to directly show the logical structures of the deeper-nested components without needing to consider the higher-level parent components.

6.1.2.2 In-Place expansion.

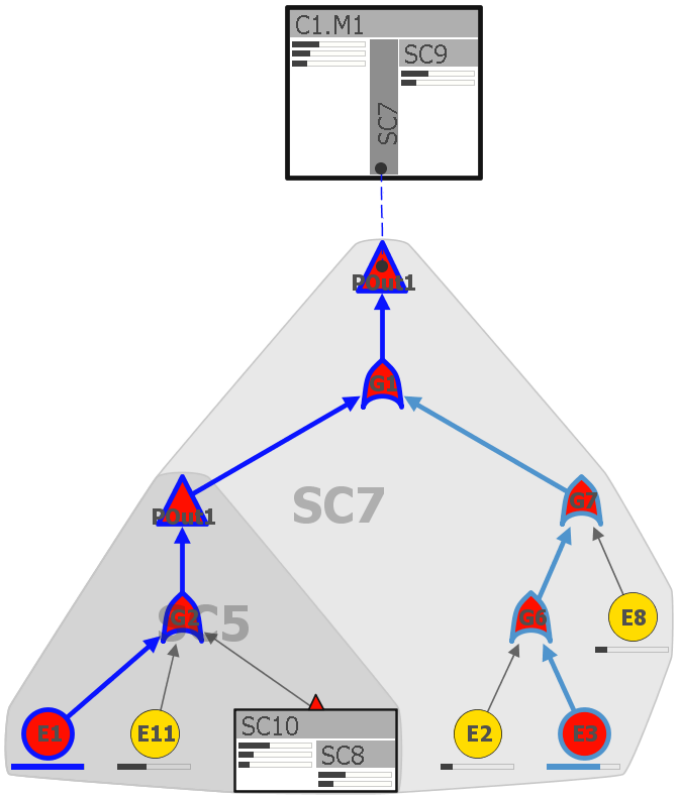
The in-place expansion implements case 4 in Table 4.7. This way, the architectural view of a CFT component is simply replaced with the corresponding logical structure. The benefit is that engineers can obtain the continuous logical failure flow. For example, Figure 6.3 (c) shows the continuous failure flow of the basic event "E1" (highlighted by blue border). In contrast, the ordinary representation concept



(a) Architectural view.



(b) Referencing expansion.



(c) In-place expansion.

Figure 6.3: Expansion concepts. The importance bar and critical path of the most important basic event "E1" are highlighted in blue. The importance bar and critical path of the second important basic event "E3" are highlighted in light blue. (a) Architectural view. The blue importance bar represents the most important basic event and the light blue bar represent the second most important basic event. (b) Showing structure of CFT component "SC7" using referencing expansion concept. A gray blob indicates the scope of the CFT structure. (c) Showing structure of the sub-CFT component "SC5" using in-place expansion concept. The continuous critical path is displayed.

separates the CFT component structure over independent views. It is difficult to represent the continuous paths of failure flow.

6.1.3 Highlighting Methods

In order to represent the failure propagation of the important basic events, we provide two highlighting mechanisms: highlighting regarding the important basic events automatically determined by a criterion, and highlighting with respect to the specific basic event.

6.1.3.1 Highlighting the important Basic Events

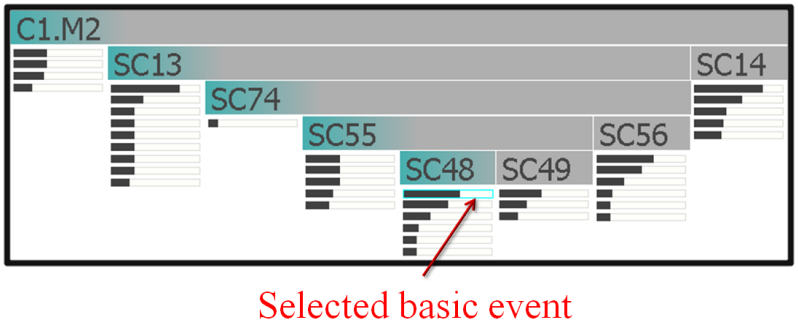
The important basic events may be automatically determined by a two-level estimation criterion: the most important basic event(s) and the secondary level important basic events. Engineers are allowed to configure the thresholds of the criterion. We use colors to highlight the borders of the important basic events and their critical paths. In order to avoid the conflict with the colors of unreliability levels, we select blue (for the most important basic events), light blue (for the second-level important basic events), and black (for the unimportant basic events). Figure 6.3 (a) shows the highlighting of the importance bars, and Figure 6.3 (b) and (c) represent the path highlighting.

6.1.3.2 Highlighting the specific Basic Event

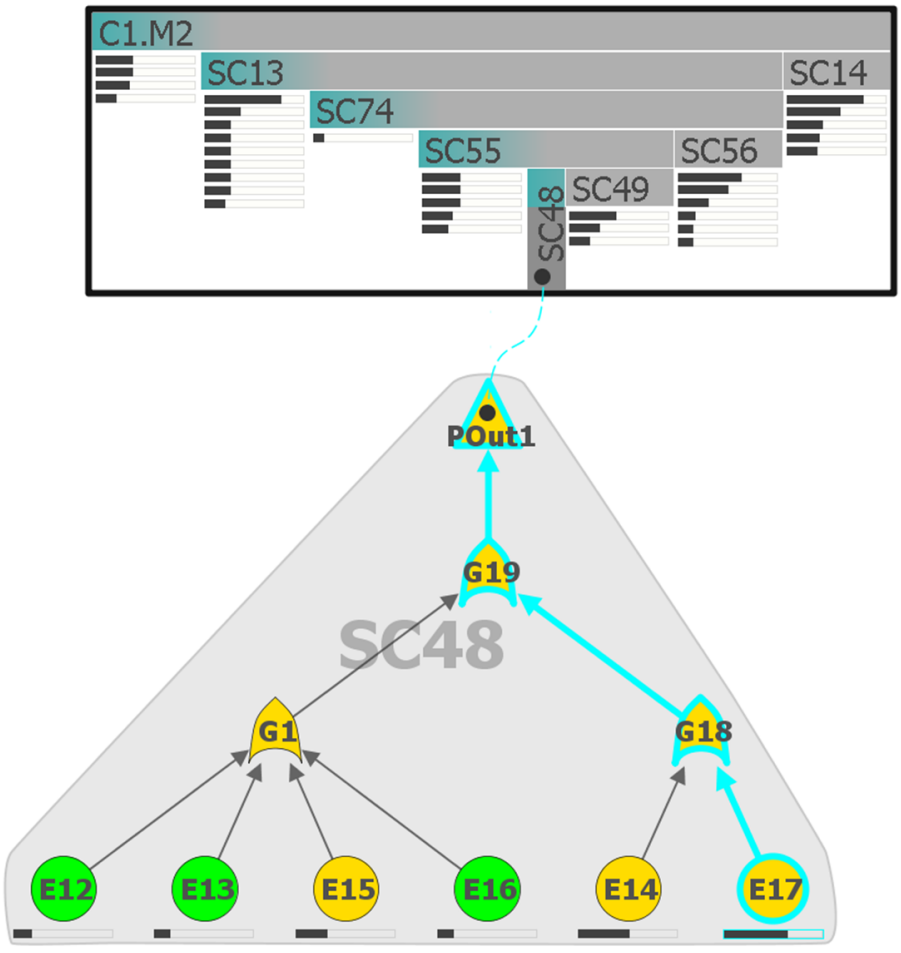
A CFT component may be influenced either by the failure of the nested critical sub-CFT components or by the input failure. When selecting a basic event in the architectural view, the influenced components are highlighted by using gradient cyan areas on the left side of the rectangles (Figure 6.4 (a)). This represents the influence along the nesting structure of components. This highlighting concept is useful for analyzing the specific important basic event with respect to not only the influenced CFT components, but also the logical faliure flow. Figure 6.4 (a) shows an example. We select a basic event belonging to the CFT component “SC48”, and the cyan areas indicate the influenced CFT components: “SC48”, “SC55”, “SC74”, “SC13”, and “C1.M2”. Additionally, when showing the logical structures of the parent CFT components of the selected basic event, the critical path of this basic event will be highlighted by a cyan border (Figure 6.4 (b)). In this way, for a selected basic event, engineers may investigate the logical failure in particular parent components while maintaining the overview of the influenced components in the architectural view.

6.1.4 Adapting CFT Structure

For representing the failure flow with respect to the important basic events, we visualize the logical structures of CFT components (i.e., the CFT structures) using a hierarchical network based on the concept of the Sugiyama algorithm [183].

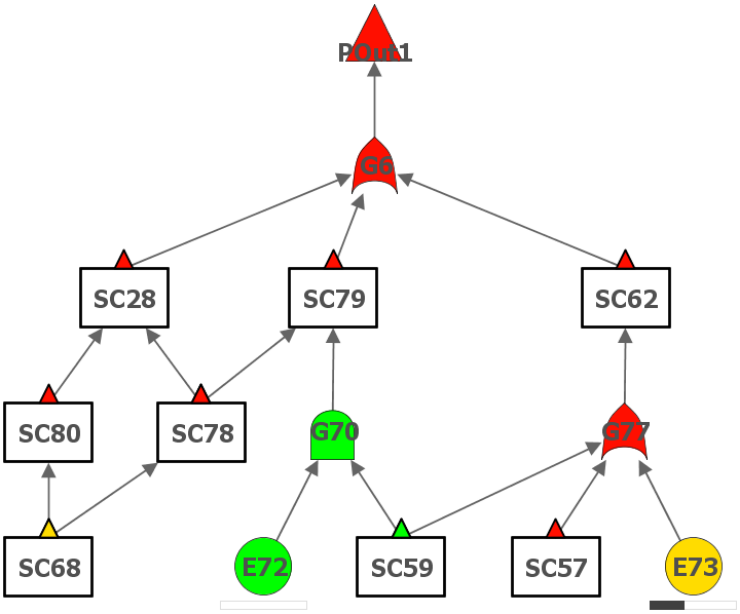


(a) Highlighting the influenced components in the architectural view using cyan areas.

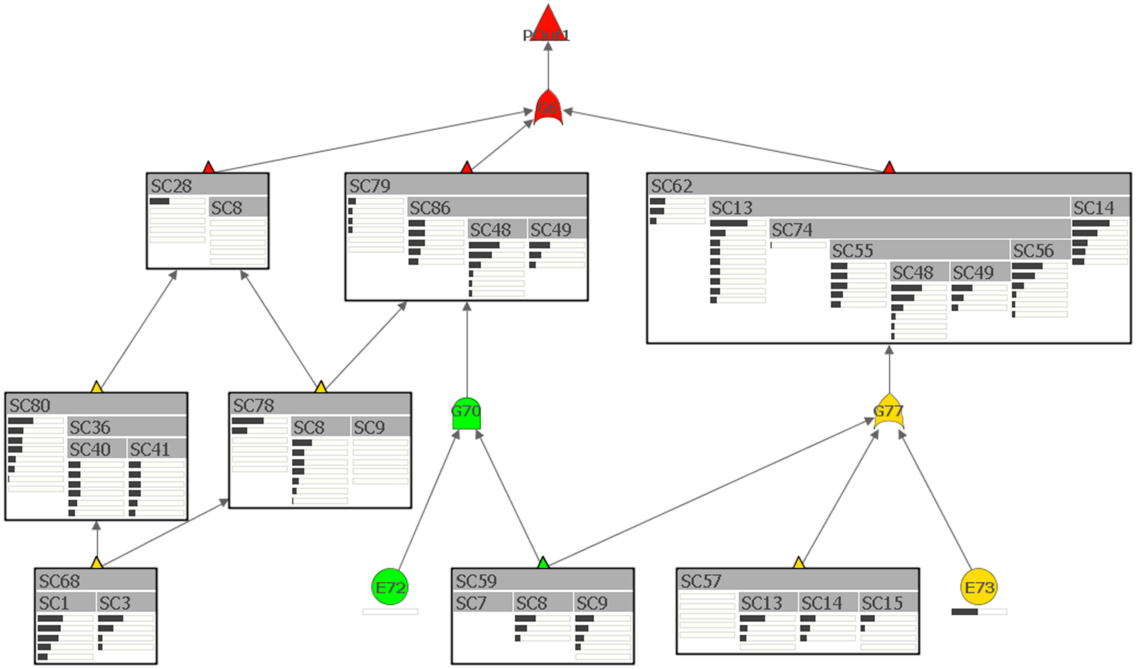


(b) Highlighting the critical path.

Figure 6.4: Highlighting the specific basic event. The selected basic event is highlighted by a cyan border.



(a) Ordinary representation for the logical structure of the system-level CFT "C1.M1".



(b) The logical structure with embedded architectural views.

Figure 6.5: Embedding architectural views into the CFT structure.

6.1.4.1 Visual Features of the CFT Structure

The failure probability of CFT elements is a meaningful context to analyze the quantitative failure propagation. We visualize the unreliability level of the nodes of the CFT structure using colors defined in section 5.1.2. We visualize the ordinary nodes with the US style (Figure 9.1). A sub-component is symbolically represented as a box that was defined in [108]. One or more in- and out-ports of the sub-component are represented as triangles. The architectural views of the sub-components may be optionally embedded in the logical structure instead of the original boxes (Figure 6.5). The architectural views provide more meaningful information and the boxes require less display space. The importance bars are respectively put under the basic event nodes.

We use a translucent gray blob to indicate the scope of the logical structure of a CFT component (Figure 6.6). Previously, Collins et al. [47] proposed an approach for drawing contours for representing the dataset relations. Simonetto et al. [176] proposed an algorithm for drawing translucent blobs in order to identify the overlapping datasets. Smith et al. [178] used blobs to represent the molecule compartment hierarchies in biochemical domain. Byelas et al. [33] proposed a concept using textured blobs to represent metrics for UML diagrams. Elmqvist et al. [62] have surveyed the concepts that used blobs to represent hierarchies of data.

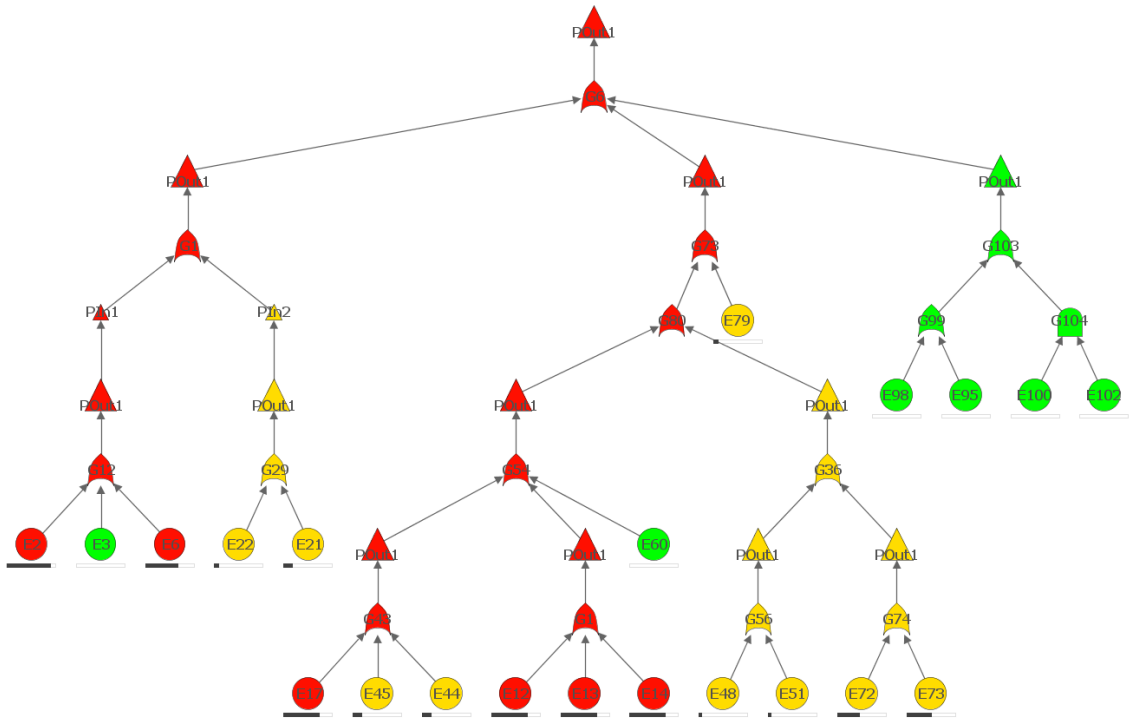
Because our blobs are translucent, the nesting of CFT components can be intuitively represented. For example, in Figure 6.6, the blobs present a multi-level nesting structure. Additionally, the ID and short description of the CFT component may be optionally displayed on the blob in order to present the basic information.

Engineers are allowed to adjust the parameters of the blob contour for adapting to the irregular shape of the CFT structure. This is helpful for addressing the overlapping issue. Using the simple convex contour engineers may quickly identify the scope of the components while using the complex multi-sided polygon contour, engineers may exactly identify that information (e.g., Figure 6.7). The method of drawing blobs is based on the algorithm proposed in [47] that is suitable to draw the complex contours for different sub-graphs in the node-link diagram.

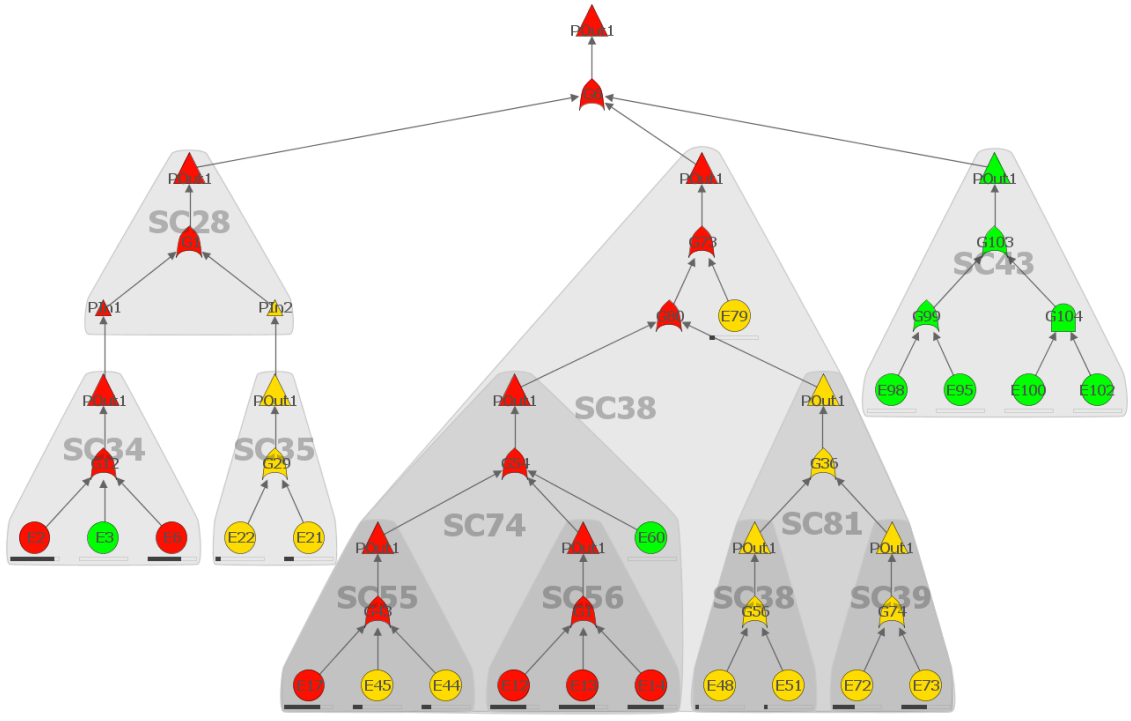
6.1.4.2 Alignment of Basic Events

There are few studies that concentrate on enhancing the arrangement of nodes of the FT/CFT structure. Usually, the ordinary concepts provide the unorganized layout for the FT/CFT structure. Besides the default arrangement of the basic event nodes (Figure 6.8 (a)), we propose two additional alignment concepts for basic event nodes in order to support the comparisons:

- Default alignment layout. CFT nodes are arranged depending on the initial hierarchical network drawing algorithm. This concept requires a small amount of display space. This layout helps to identify the distance (i.e., the number of levels in the tree) between the top event and a basic event while maintaining an efficient use of space.
- Local alignment layout (Figure 6.8 (b)). To facilitate the comparison of basic events inside the individual CFT component, we propose a concept, where

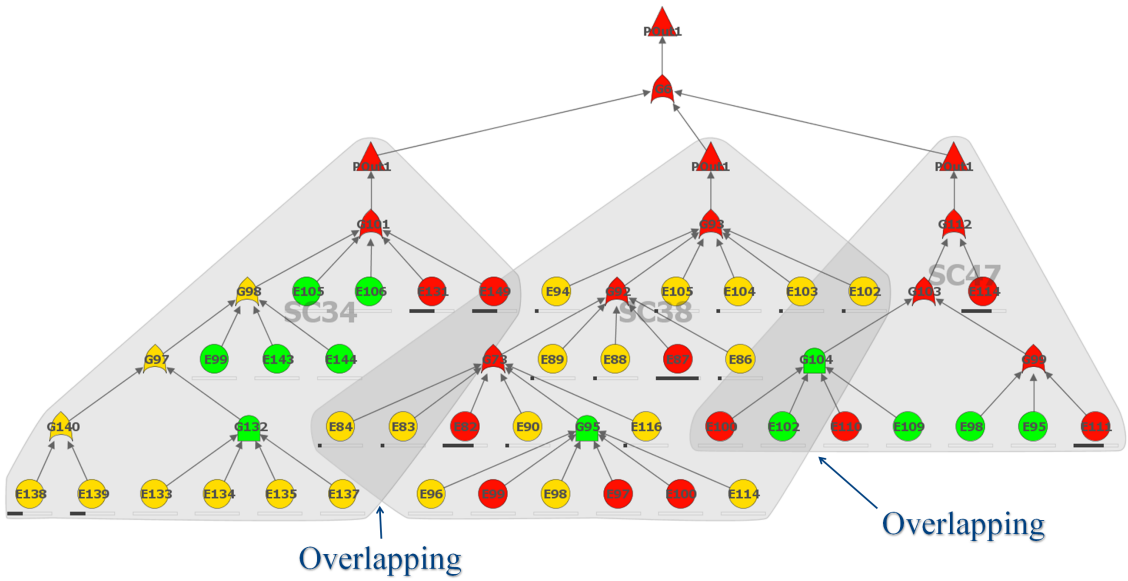


(a) CFT structure without using blobs.

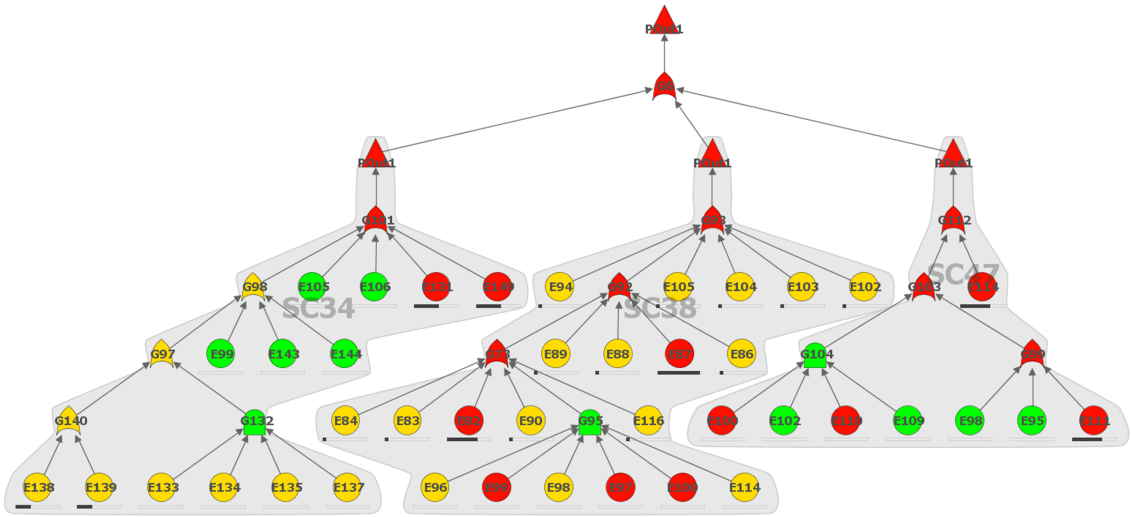


(b) CFT structure using blobs.

Figure 6.6: Blobs for the CFT components. Blobs indicate the scopes of the logical structures of CFT component. The nesting of CFT components are identified by the translucency of blobs. Brief descriptions or IDs of the components are may be printed on the blobs.



(a) Simple convex contour.



(b) Complex multi-sided polygon contour.

Figure 6.7: Contour of blobs.

aligning the basic event nodes at the bottom of the logical structure of each CFT component.

- Global alignment layout (Figure 6.8 (c)). To simplify the comparison of basic events across CFT components, we developed a global comparison layout. All basic event nodes are aligned at the bottom of the overall CFT structure. Using this layout, engineers may quickly scan the horizontal list of the basic events without the need to locate events within the complex CFT structure.

6.1.5 Importance Plot

Regarding the CFT structure, the comparison among the large number of basic events is not effective. For addressing this issue, we propose a linked view called the *importance plot* that is adjacent to the CFT view (lower part of Figure 6.15). Basic events in the importance plot are represented as circle icons and filled with the same color as that in the CFT structure. In the importance plot, a basic event is shown by its current horizontal position on the CFT structure along the x-axis and its importance along the y-axis. The importance plot represents the distribution of basic events over both the importance value and the position in the CFT structure. The rectangle icon on the plot represents the sub-component node symbolically represented in the CFT structure (i.e., the unexpanded CFT component). The vertical position of the rectangle icon depends on the maximal importance value of its basic events. In this way, engineers may quickly determine whether an important basic event exists in the component. Another significant use of the importance plot is to support the highlighting of the important basic events. Engineers may dynamically set the position of a horizontal line in the importance plot after identifying the important basic events. The nodes above the line are considered as the important basic events and their critical paths are automatically highlighted.

6.2 Application Scenarios

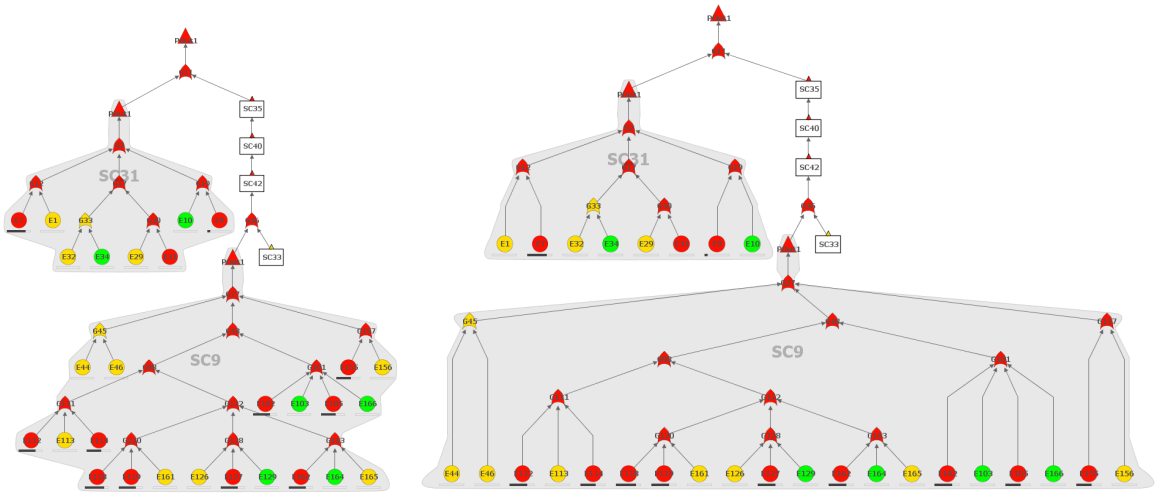
In order to present the use of our visualization, application examples are presented in this section.

6.2.1 Dataset and Configuration

We apply a CFT model that contains 1031 basic events and 45 CFT components in the examples. The model was generated depending on the safety analysis for the robot RAVON [164, 201]. The unreliability levels are defined as follows:

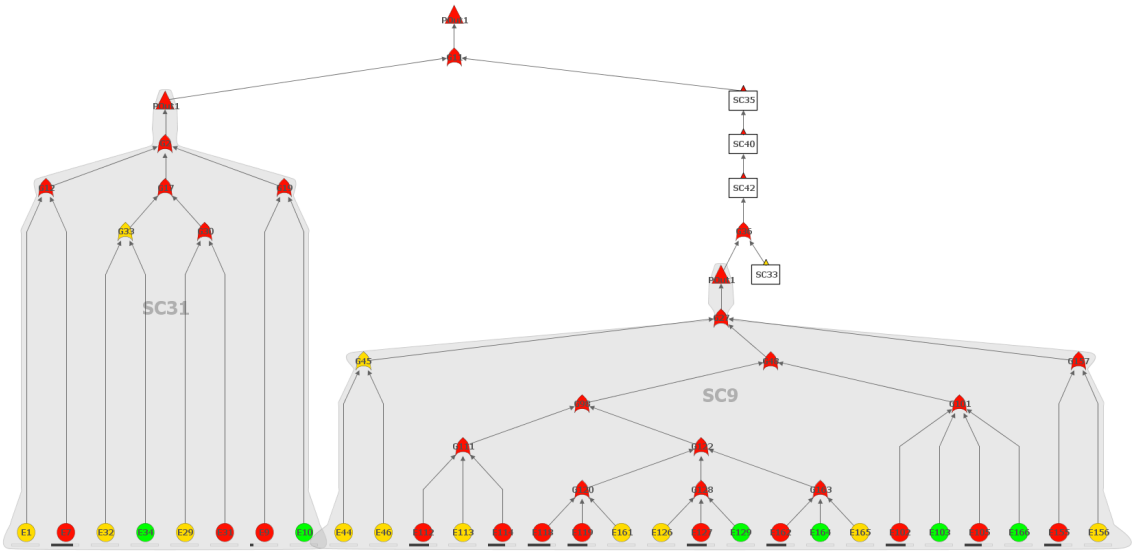
- critical level (red): (1e-2, 1]
- moderate level (yellow): (1e-4, 1e-2]
- acceptable level (green): (0, 1e-4]

The lower and upper bounds of the importance bar are respectively set to “1e-4” and “1”. In order to effectively represent the importance values in this interval, we use the logarithmic scale for the importance bars and the importance plot.



(a) Default alignment concept.

(b) Local alignment concept.



(c) Global alignment concept.

Figure 6.8: Basic event alignment layouts. The layouts offer trade offs between space compactness and ease of comparison of the basic events.

6.2.2 Example 1

A significant task of the importance analysis of the CFT is to identify the CFT components influenced by the important basic events. The goal of this example is to identify the CFT components influenced by the most important basic event and to understand how the components are influenced.

6.2.2.1 Analysis Process

The goal is achieved with the following steps:

1. We first identify the most important basic event and the most critical component that contains. As a result, Figure 6.9 shows that the obviously important basic events are distributed in seven CFT components: "SC13", "SC49", "SC28", "SC114", "SC19", "SC53", and "SC45". These are the critical CFT components. The CFT Component "SC49" contains the most important basic event. Thus, the "SC49" is the most critical component and needs to be primarily addressed.
2. We identify the CFT components influenced by the most important basic event. By selecting the most important basic event in the CFT component "SC49", the influenced CFT components are highlighted by small cyan areas on the left side of the rectangles.

   Figure 6.10 shows the analysis results. The parent CFT component "SC49" of the most important basic event is certainly influenced. The components "C1.M1", "SC62", "SC13", "SC74" and "SC55" are influenced because they directly or indirectly contain the most critical CFT component "SC49". The failure of "SC49" is transferred along the nesting structure. On the other hand, we cannot effectively understand how components "SC73" and "SC12" are influenced that do not have nesting relations with the most critical CFT component "SC49". Even, the CFT component "SC12" is the sub-component of "SC49". Thus, we need to investigate the corresponding failure flow along the logical structure of the CFT.
3. We investigate the influence by the logical failure flow. In order to understand the influence on the CFT component "SC73", we display the logical structure of its parent CFT component "SC74' using the referencing expansion. The result is shown in Figure 6.11 and Figure 6.12. With the help of the highlighted paths, we identify that the failure coming from the CFT component "SC55", which indirectly contains the critical CFT component "SC49", flows into "SC73" by going through gate "G7". This explains how the CFT component "SC73" is influenced.

   Then we investigate the influence on the CFT component "SC12". Because "SC12" is the sub-component of "SC49", we need to show the logical structure of "SC49" (Figure 6.13). The failure of the most important basic event "E2" flows into the sub-component "SC12". This is why "SC12" is one of the influenced components.

In the analysis process, we optionally show the (sub-)architectural views of the sub components according to the requirements. Only if a (sub-)architectural view

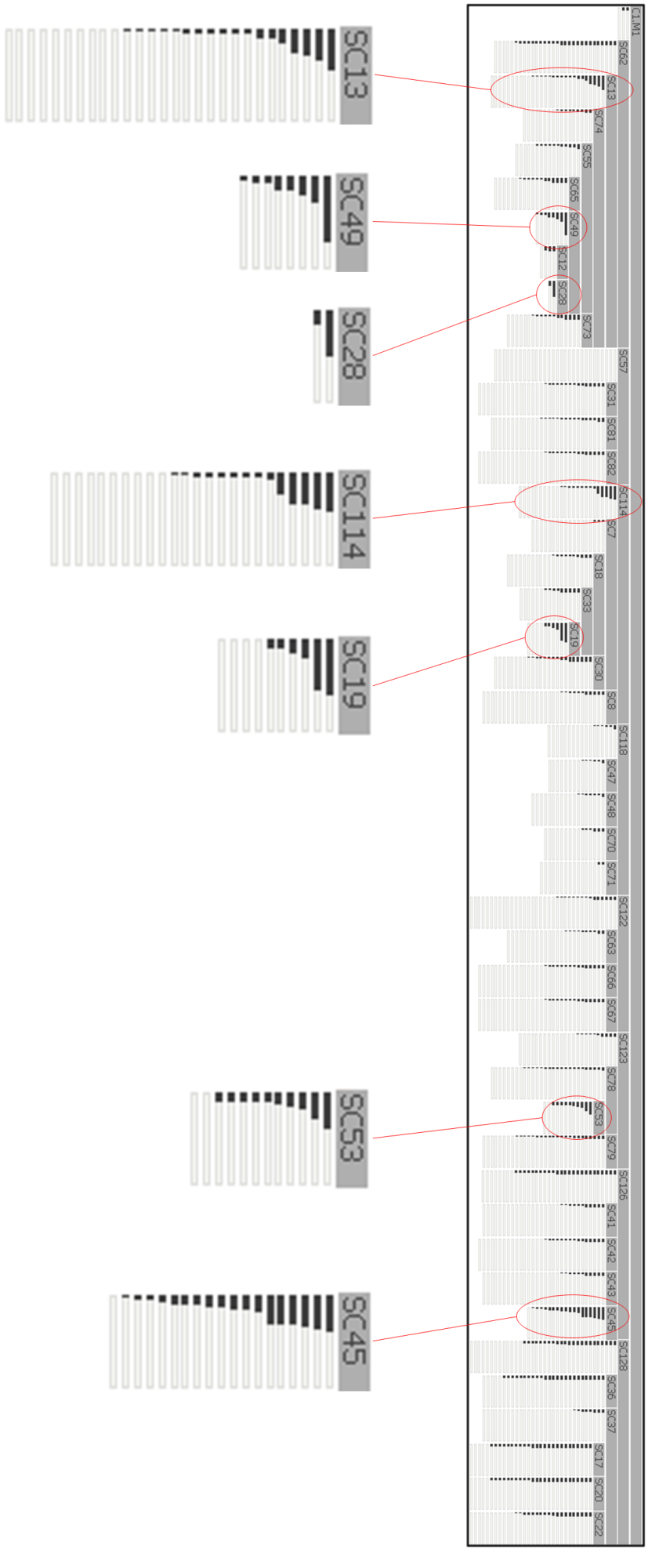


Figure 6.9: Application example 1. Architectural view of the system level CFT component.

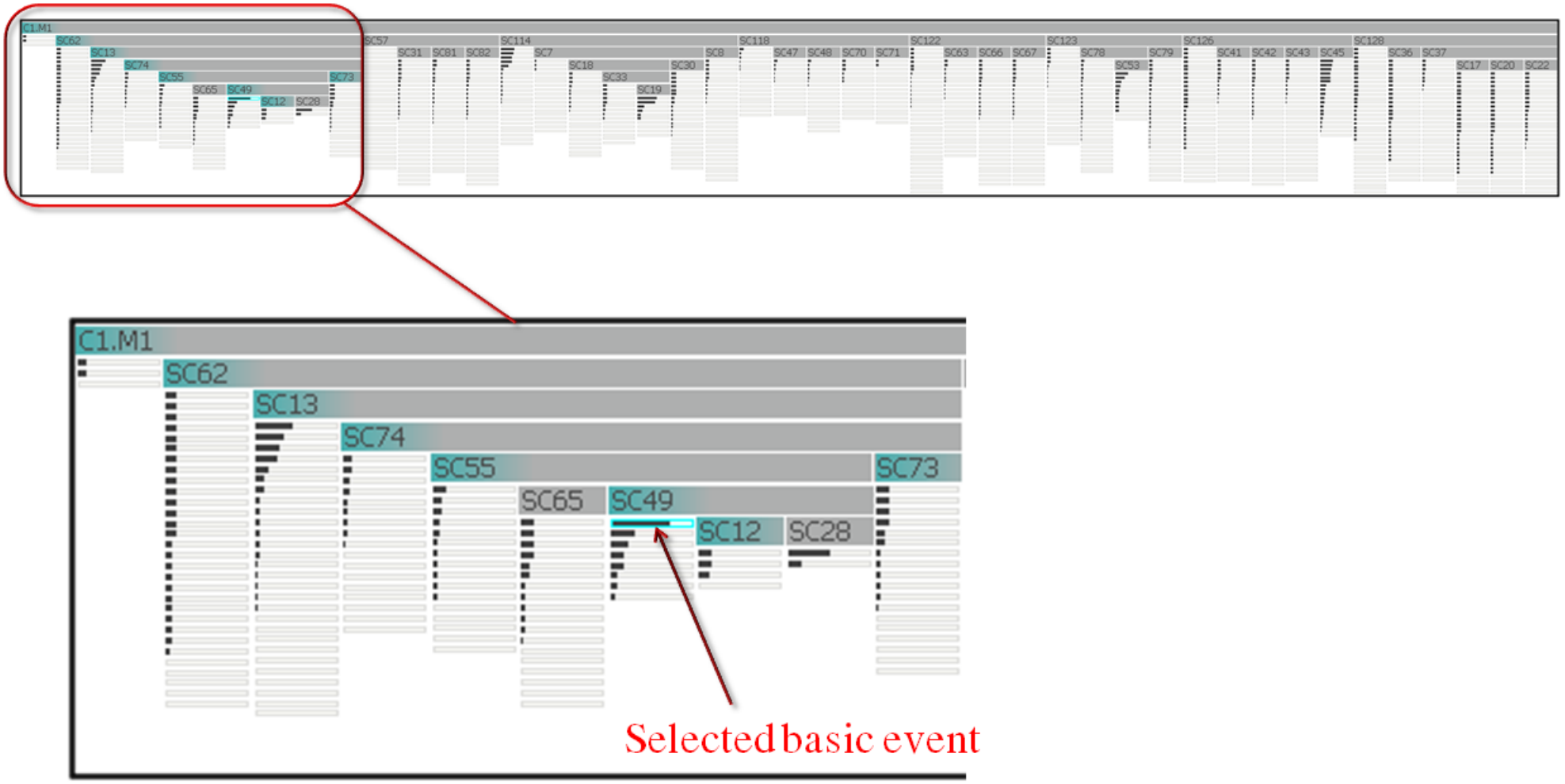


Figure 6.10: Application example 1. The CFT components influenced by the most important basic event are highlighted.

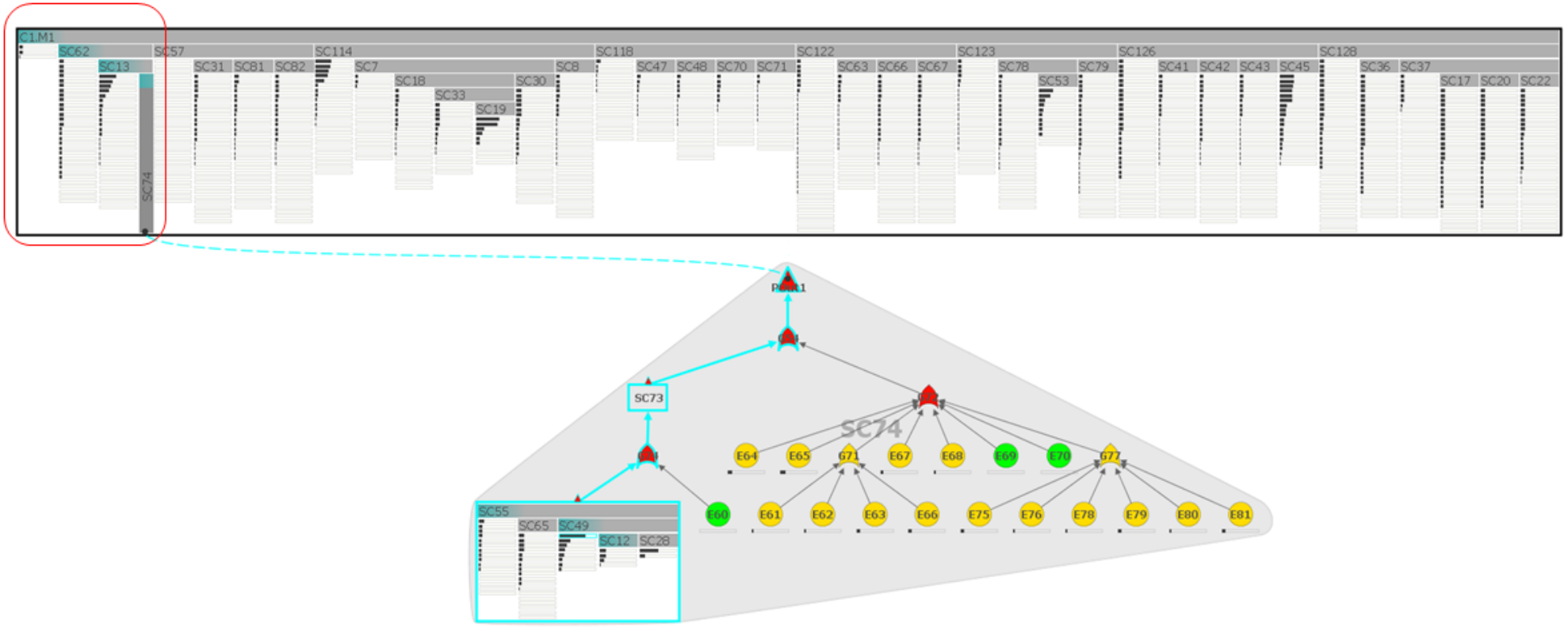


Figure 6.11: Application example 1. The CFT structure of the CFT component "SC74". A partially enlarged view is shown in Figure 6.12.

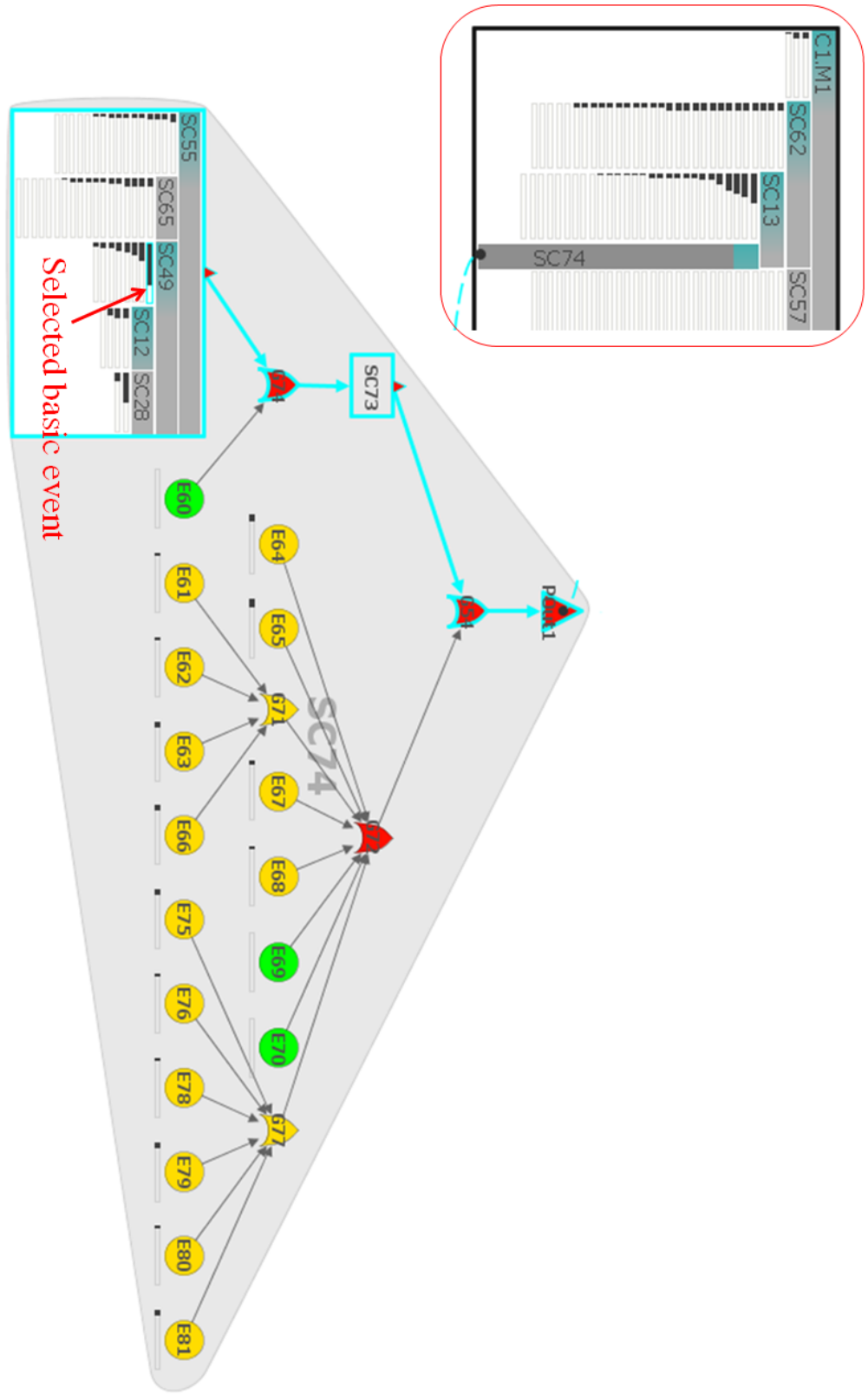


Figure 6.12: Application example 1. A partially enlarged view of Figure 6.11.

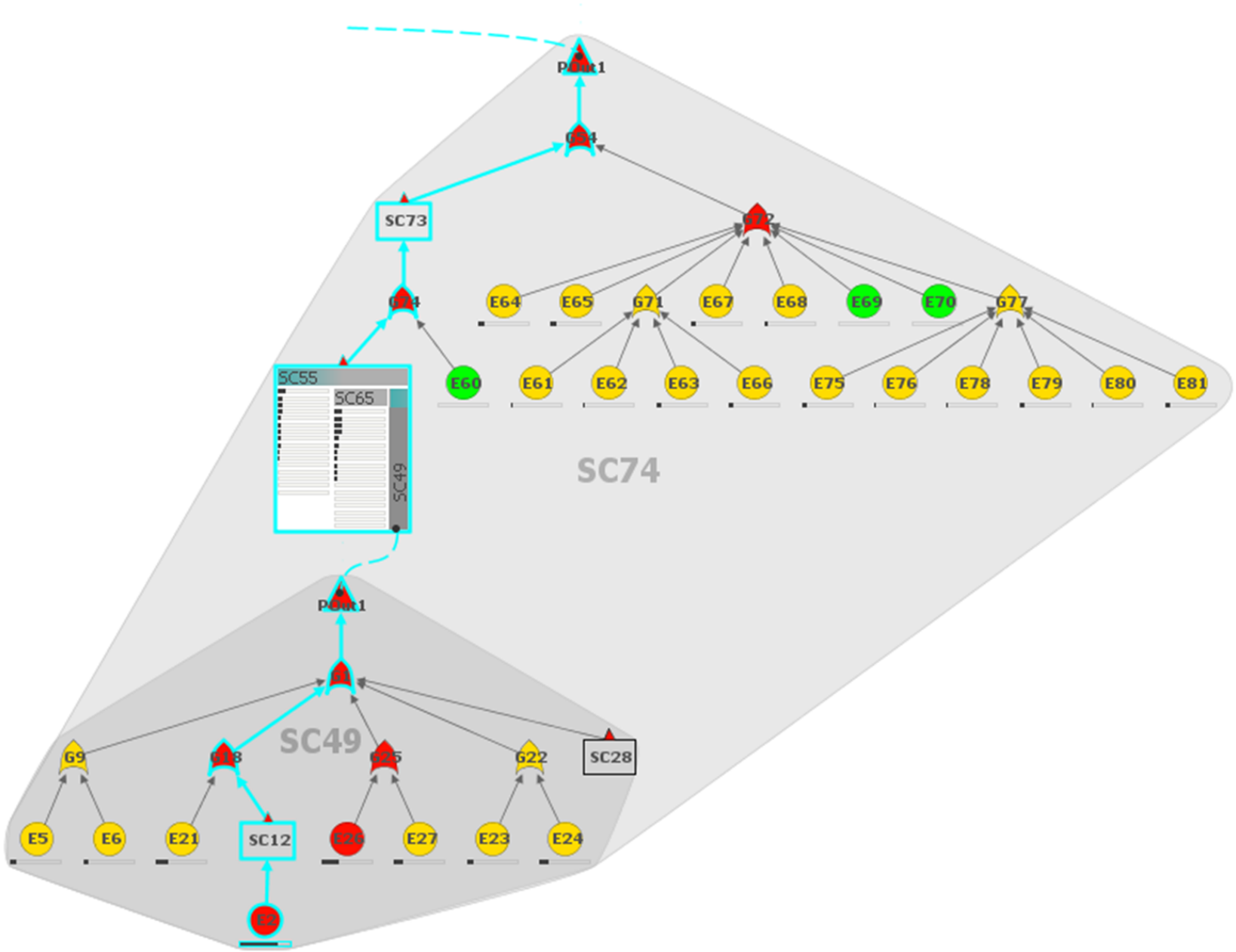


Figure 6.13: Application example 1. The logical structure of the CFT component “SC49”.

provides meaningful information, do we display it instead of the simple rectangular symbol, e.g., "SC55" in Figure 6.12. Otherwise, we prefer the simple rectangle because of its small space requirement, e.g., "SC73" in Figure 6.12. This example depicts how to identify the significant components and how to understand the failure propagation using our visualization.

When using the ordinary concept, it is difficult for us to obtain a pattern with respect to the distribution of the important basic events over the hierarchical system model. When identifying the influenced CFT components, we have to go through all higher-level parent components step by step. Additionally, using the traditional concepts, we cannot analyze the logical failure flow while maintaining the overview of the importance of basic events.

6.2.3 Example 2

An overview of multiple critical paths brings meaningful content for understanding the complex failure propagation along the logical structure. For example, critical paths of two important basic events are logically connected by an AND-gate. This facilitates to identify the logical function with respect to the important basic events. This example shows the analysis of multiple failure flow of the important basic events. The objective of the example is to identify and analyze multiple critical paths.

6.2.3.1 Analysis Process

The objective can be achieved by performing the following steps:

1. We identify the critical CFT components (Figure 6.9). The architectural view clearly shows the resulting components: "SC13", "SC49", "SC28", "SC114", "SC19", "SC53", and "SC45".
2. We show the logical structures of the critical components. Figure 6.14 shows that "SC49" is included in "SC13". Because the structure of the CFT component "SC13" does not provide meaningful information for the goal, we only display the structure of "SC49" without showing the structure of the parent component (Figure 6.16). In the same way, the structure of the CFT component "SC19" is displayed (Figure 6.17).
3. We determine the important basic events using the importance plot (lower part in Figure 6.15). According to the overview of the importance of basic events in the plot, we identify a few nodes obviously having the higher importance. We set the horizontal line between these nodes and others. Then the corresponding basic events and the critical paths are automatically highlighted in the CFT structure. The results are shown in Figure 6.16, Figure 6.17, and Figure 6.18. The blue path indicates the most important basic event and light-blue paths indicate the second most important basic events.
4. We then investigate the critical paths the most critical CFT component "SC49". We perform a top-down analysis for this component with respect to multiple failure flow. The dynamically displayed node labels in Figure 6.19

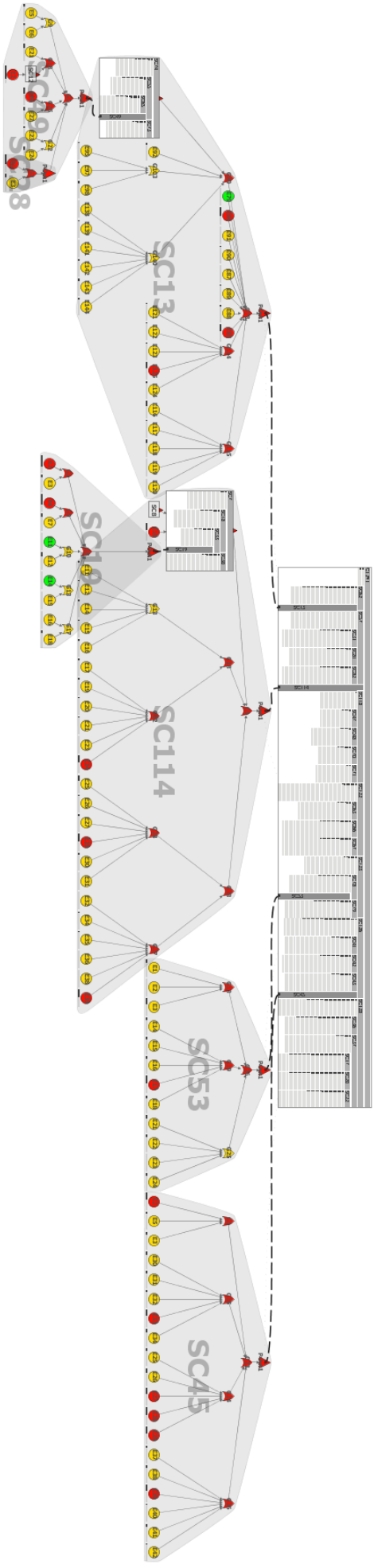


Figure 6.14: Application example 2. Logical structures of the critical CFT components.

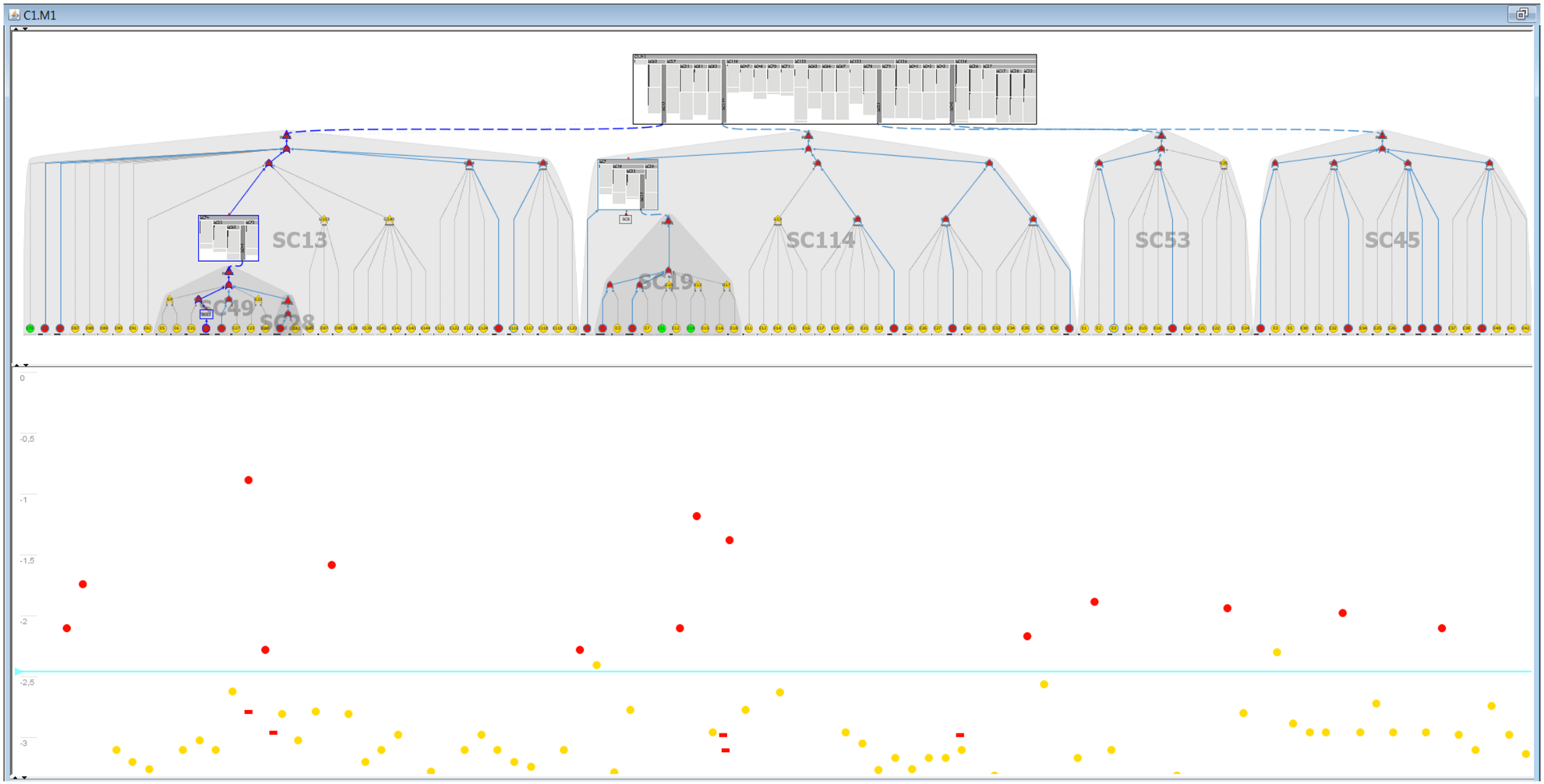


Figure 6.15: Application example 2. Highlighting multiple critical paths. The partially enlarged views are shown in Figure 6.16, Figure 6.17, and Figure 6.18.

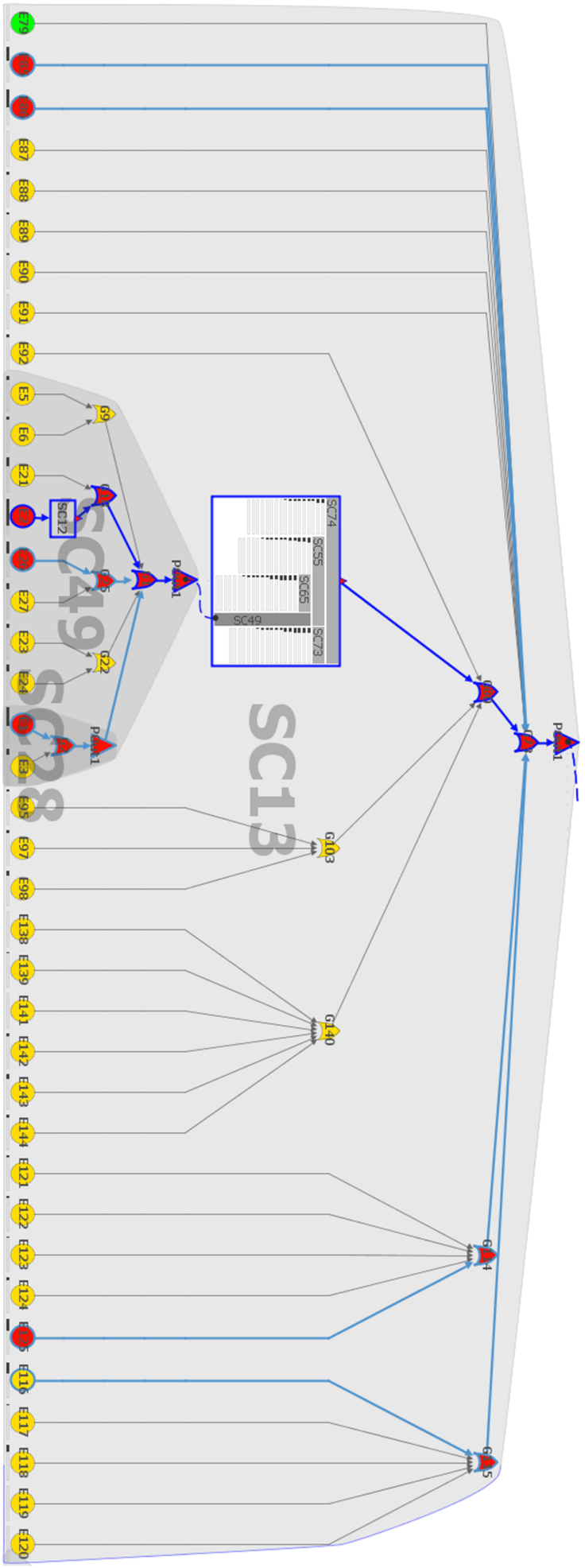


Figure 6.16: Application example 2. A partially enlarged view of the CFT component "SC13".

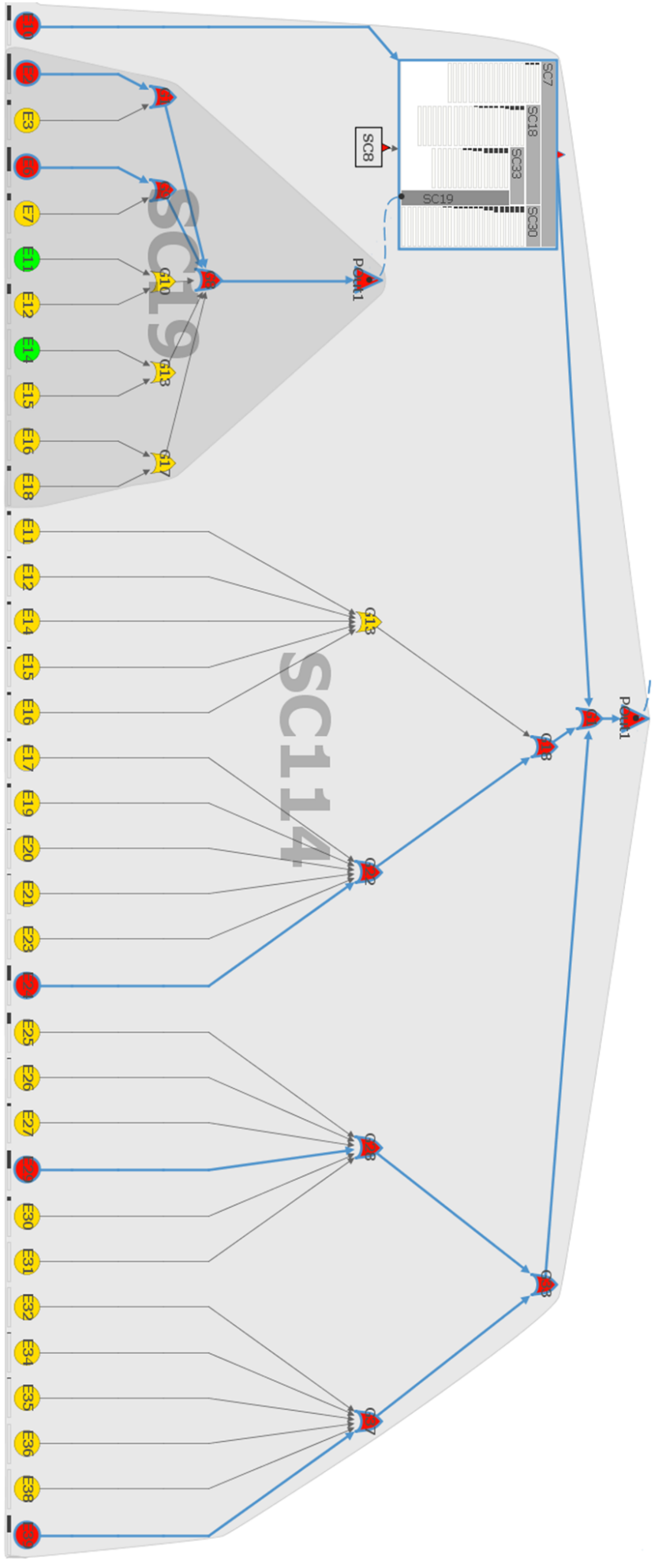


Figure 6.17: Application example 2. A partially enlarged view of the CFT component "SC114".

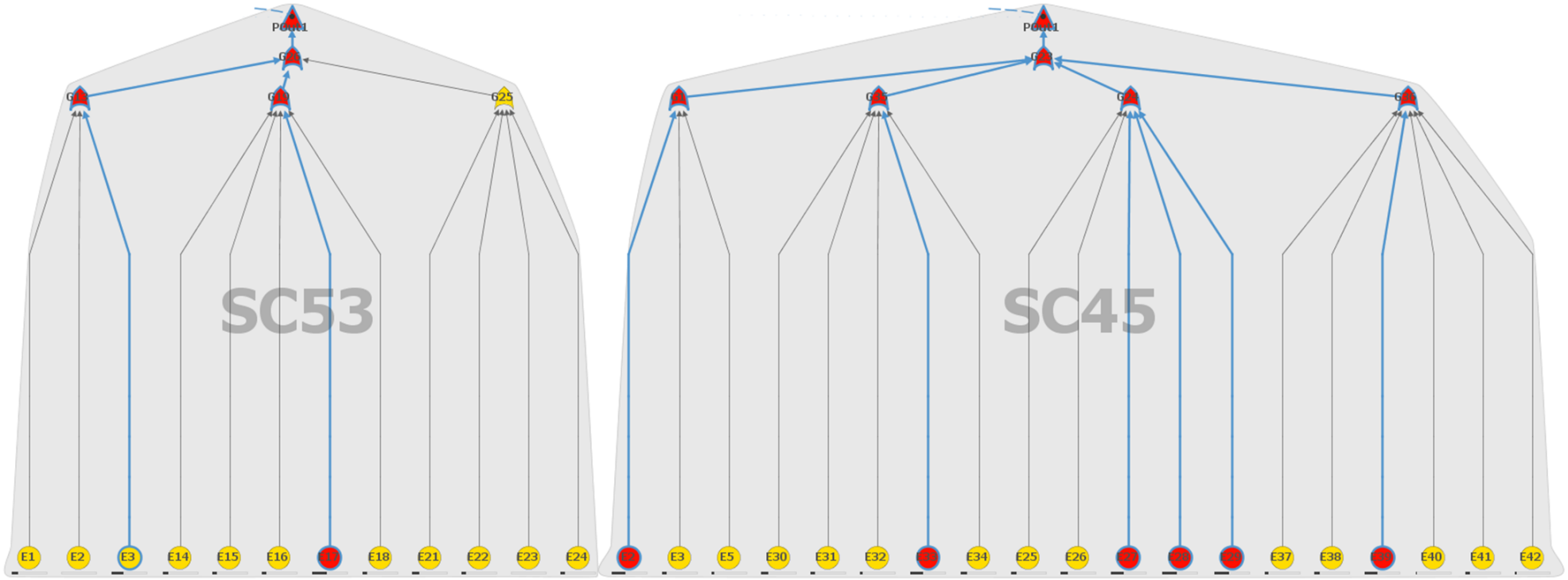


Figure 6.18: Application example 2. A partially enlarged view of components “SC53” and “SC45”.

describe the following semantic meaning of the failure propagation. This component may fail because of three highly possible failures: the intermediate failures of "G18 (DSP provides incorrect data)" (marked with a red circle), "G25 (Rotating element works incorrectly)" (marked with a red circle), and failure of the sub-component "SC28 (Scan-ID-Extractor)" (marked with a red circle).

5. We prioritize the critical paths. First we may identify that the path of the "G18" is the most critical because the path is highlighted with blue (Figure 6.19). The path of "G25" and the path inside "SC28" are the secondary level most critical because they are highlighted in light-blue. We compare the importance bars between basic events "E26" and "E1". The result shows that the failure of "G2" (referring to "E1") is more critical than the failure of "G25" (referring to"E26"). We then analyze the failure propagation as follows according to the priorities of the paths:

   - We analyze the failure of "G18 (DSP works incorrectly)". This failure may be caused by the failure of the CFT component "SC12 (DSP)" or the failure of "E21 (DSP adapter is defective)". According to the highlighted critical path, we notice that "SC12" is more risky.

     We then focus on the CFT component "SC12" that represents the DSP. We show the logical structure of this component and investigate the internal failure flow (Failure 6.20). The highlighted critical path shows that the incorrect data of DSP is generated most likely by the incorrect pivoting angle ("G4"). The incorrect angle is most likely caused by the input failure that comes from the basic event "E2 (Distance sensor is defective)". Thus, this result explains that the incorrect data coming from DSP ("SC12") is mostly due to the failure of the distance sensor ("E2") rather than the DSP itself. When improving the CFT component "SC49 (Rotating Laser Scanner)" in future, the sensor rather than the DSP needs to be focused on.
   - In order to analyze the failure coming from the CFT component "SC28 (Scan-ID-Extractor)" , we investigate the logical structure of the component. The highlighted critical path shows that the failure of the Scan-ID-Extractor is most likely caused by the basic event "E1 (Extractor chip is defective)".
   - Finally we analyze the failure of "G25 (Rotating element incorrectly works)". The highlighted path shows that this failure is most likely caused by the basic event "E26 (Rotating element is defective)" rather than the basic event "E27 (Linking part is defective)".

   Using our visualization, we may simultaneously analyze multiple highly possible failures that may trigger the CFT component "SC49 (Rotating Laser Scanner)" to occur. The highlighted and continuous critical paths may provide meaningful context information.

The example shows the effective investigation of the multiple critical CFT components and multiple critical paths by cooperating between the CFT view and the importance plot. At the beginning of the analysis, the architectural view is used

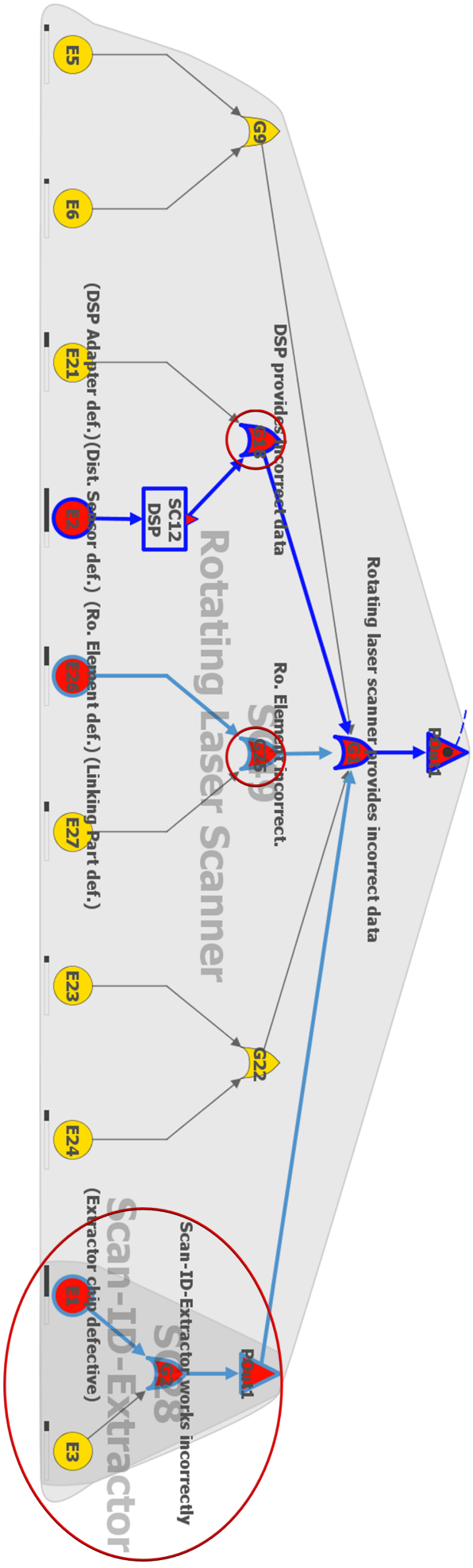


Figure 6.19: Application example 2. Analysis of multiple critical paths.

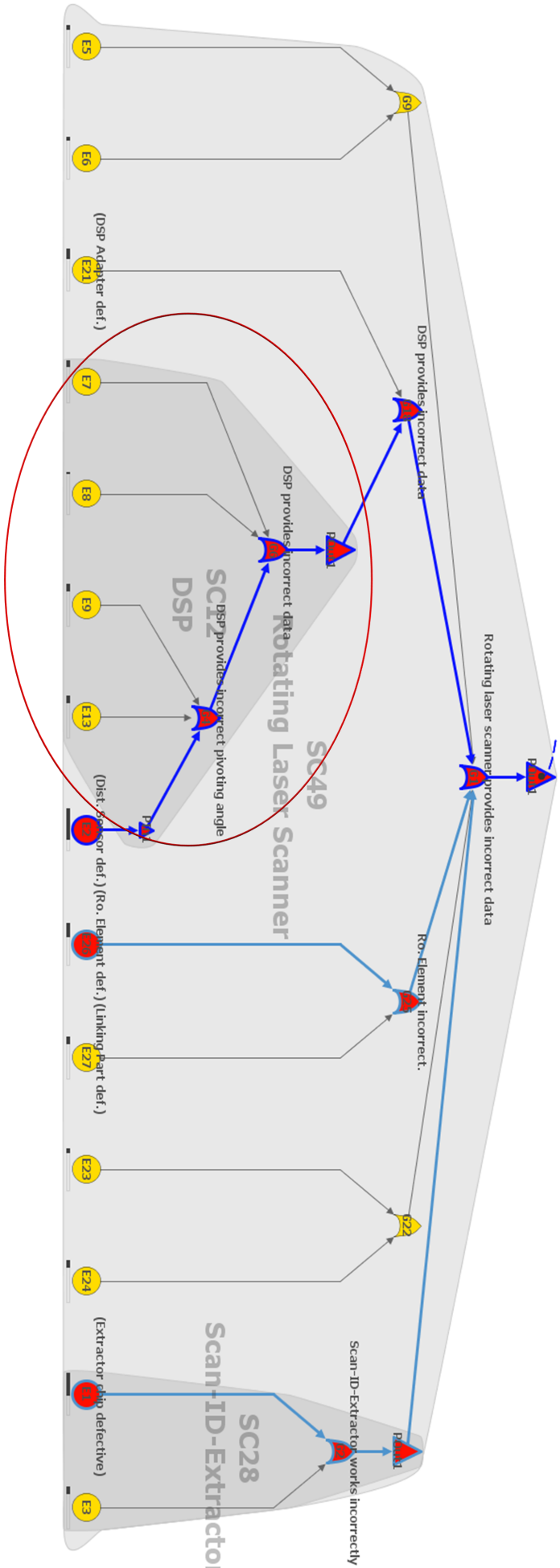


Figure 6.20: Application example 2. Critical path inside the sub component.

to quickly identify the significant CFT components that have the important basic events. Then, engineers are free to focus on the identified CFT components without wasting effort on the irrelevant components. The importance plot provides an overview for ranking the basic events belonging to the critical components. It offers support in quickly and interactively determine the multiple important basic events for identifying the critical CFT components. The highlighting of the multiple paths is helpful for the parallel analysis of the failure flow.

6.2.4 Example 3

The system-level component contains the primary (sub-)CFT components. Engineers want to investigate the overall distribution of the important basic events over the primary components while analyzing the logical structure of the system-level component. In this example, we analyze the importance of basic events with respect to the logical relations between the primary components. In addition, we focus on the failure propagation of the specific basic event.

6.2.4.1 Analysis Process

We perform the task by following steps:

1. We analyze the distribution of the important basic events over the primary components. The visible structure (Figure 6.21) shows that there are eight primary components (with labels on the top rectangles of the architectural views): “SC62”, “SC122”, “SC123”, “SC128”, “SC118”, “SC126”, “SC57”, and “SC114”. The basic events of the system-level CFT components have only low importance. The architectural views of the primary components show the distribution of the important of basic events (Figure 6.22):
   - the primary CFT component “SC62” contains critical (sub-)CFT components: “SC13”, “SC49”, and “SC28”;
   - the primary CFT component “SC123” has a critical (sub-)CFT component: “SC53”;
   - the primary CFT component “SC126” has a critical (sub-)CFT component: “SC45”;
   - the primary CFT component “SC114” itself has the important basic events and also has a critical (sub-)CFT component “SC19”.
2. We investigate the influence of the specific important basic event. As an example, we focus on the most important basic event in the CFT component “SC19”. By clicking the first importance bar that represents the most important basic event, the cyan areas of the rectangles indicate the influenced CFT components. Figure 6.23 and Figure 6.24 show the result. Inside the primary CFT component “SC114”, the failure is transferred along the nesting structure. The path is “SC19 → SC33 → SC18 → SC7 → SC114”. The outcoming failure of “SC114” flows into the primary components “SC123” and “SC128”. In the CFT component “SC128”, the (sub-)CFT component “SC36” is also influenced by the input failure. The primary CFT component “SC123” is also

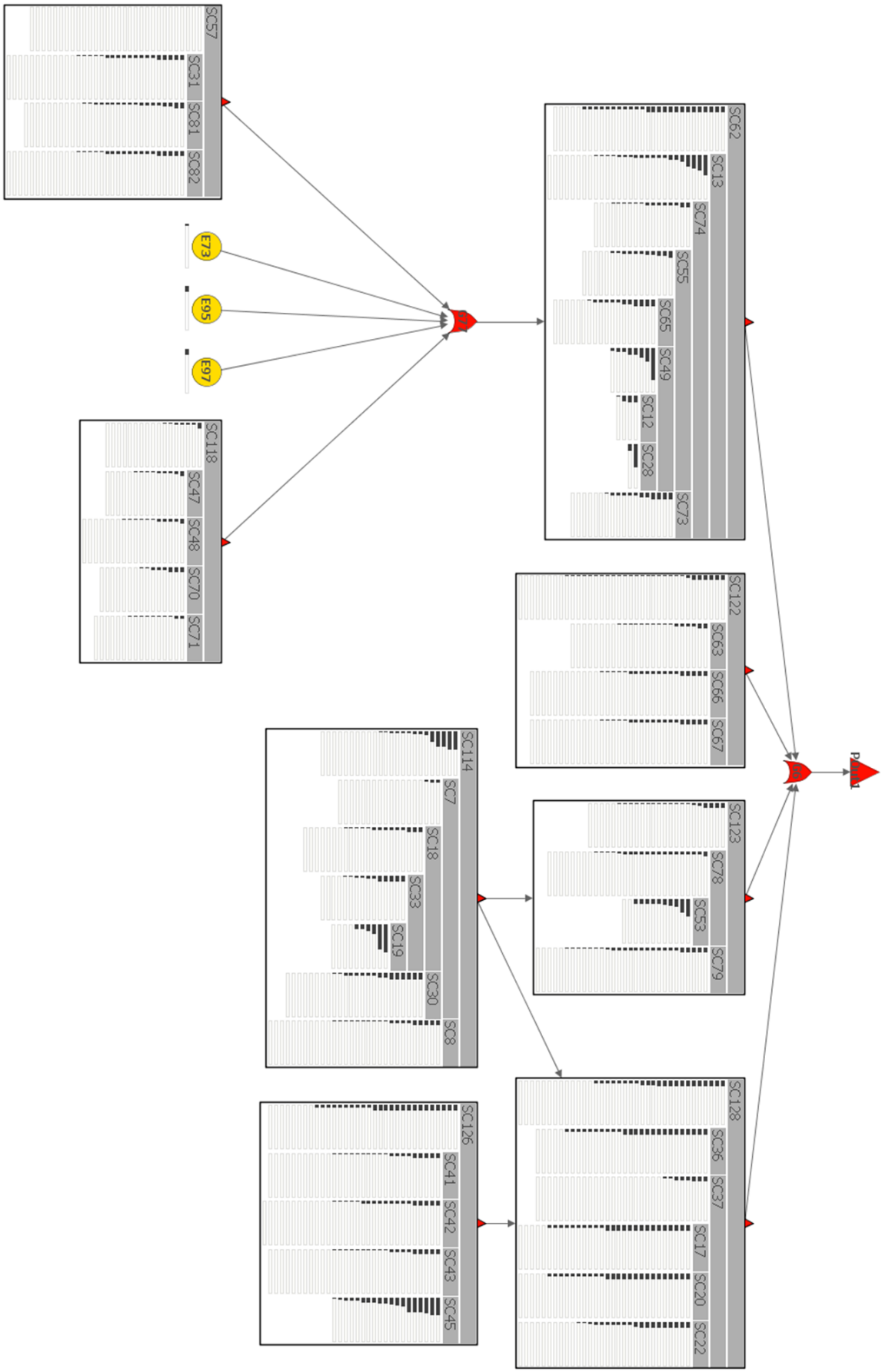


Figure 6.21: Application example 3. Analysis of importance of basic events with respect to the system-level architecture of the CFT. There are 8 primary (sub) components.

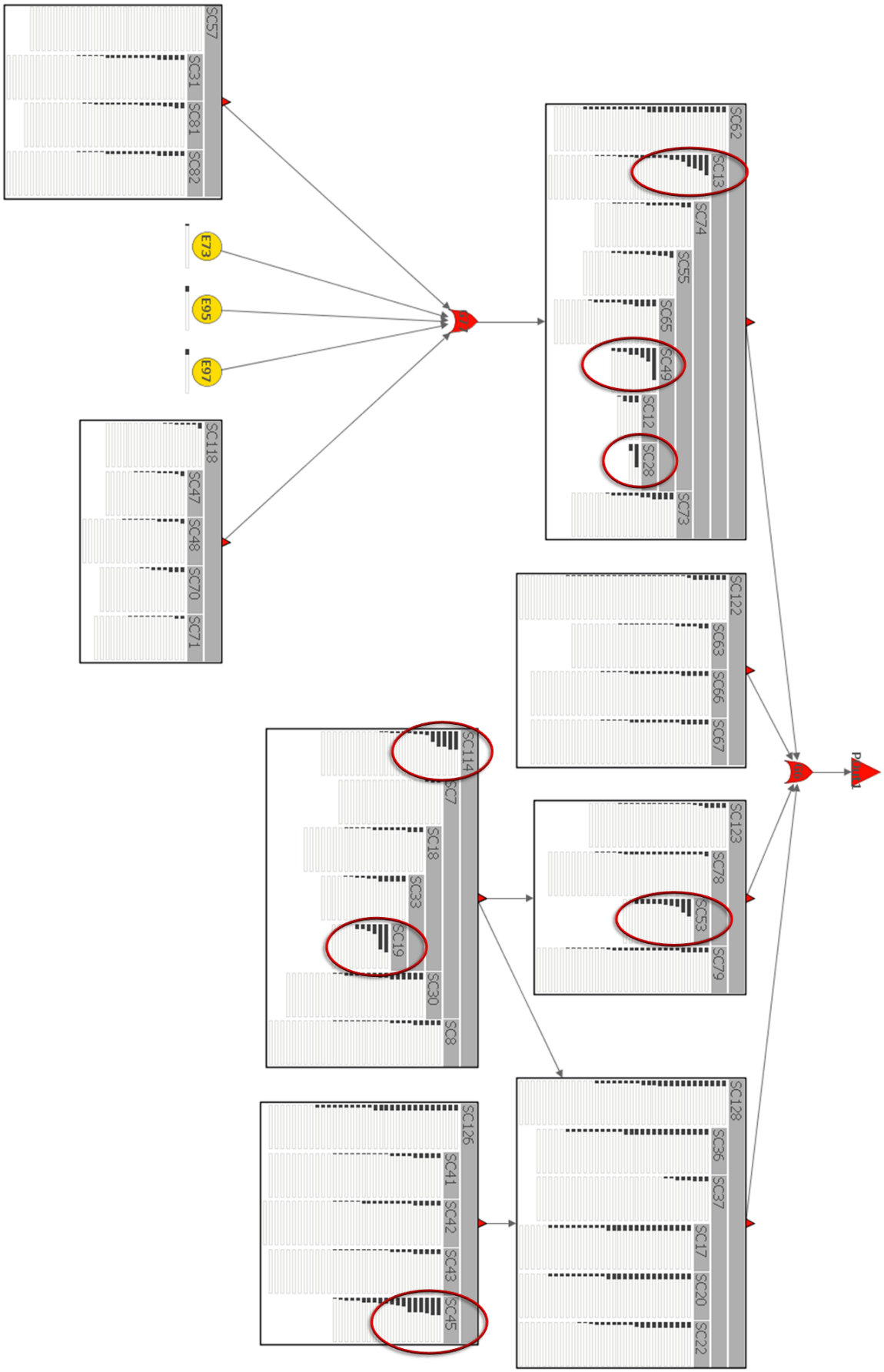


Figure 6.22: Application example 3. Critical sub-components. There are 7 critical sub components.

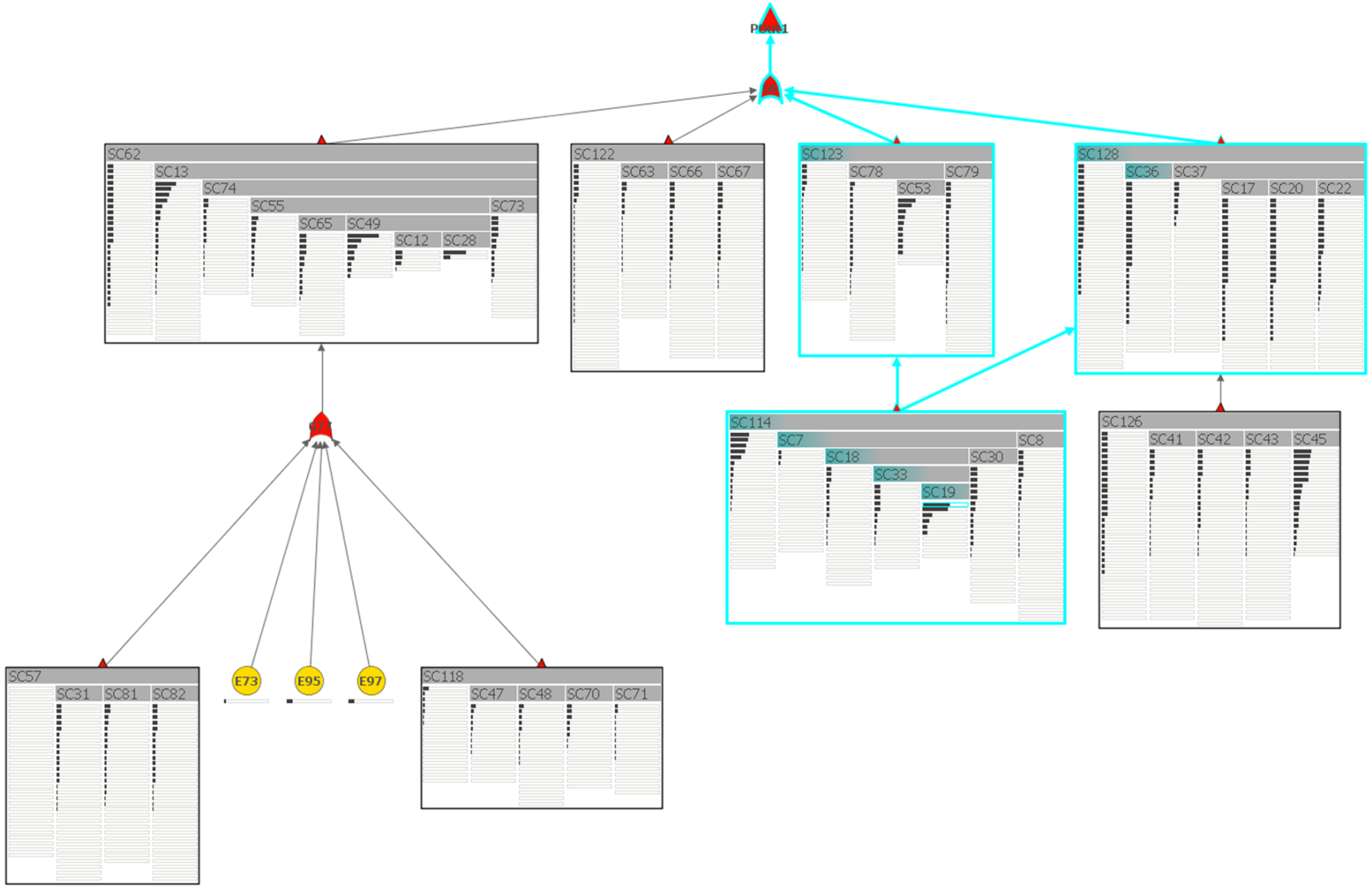


Figure 6.23: Application example 3. Influence of the most important basic event of the CFT component “SC19”. The CFT view is partially enlarged in Figure 6.21.

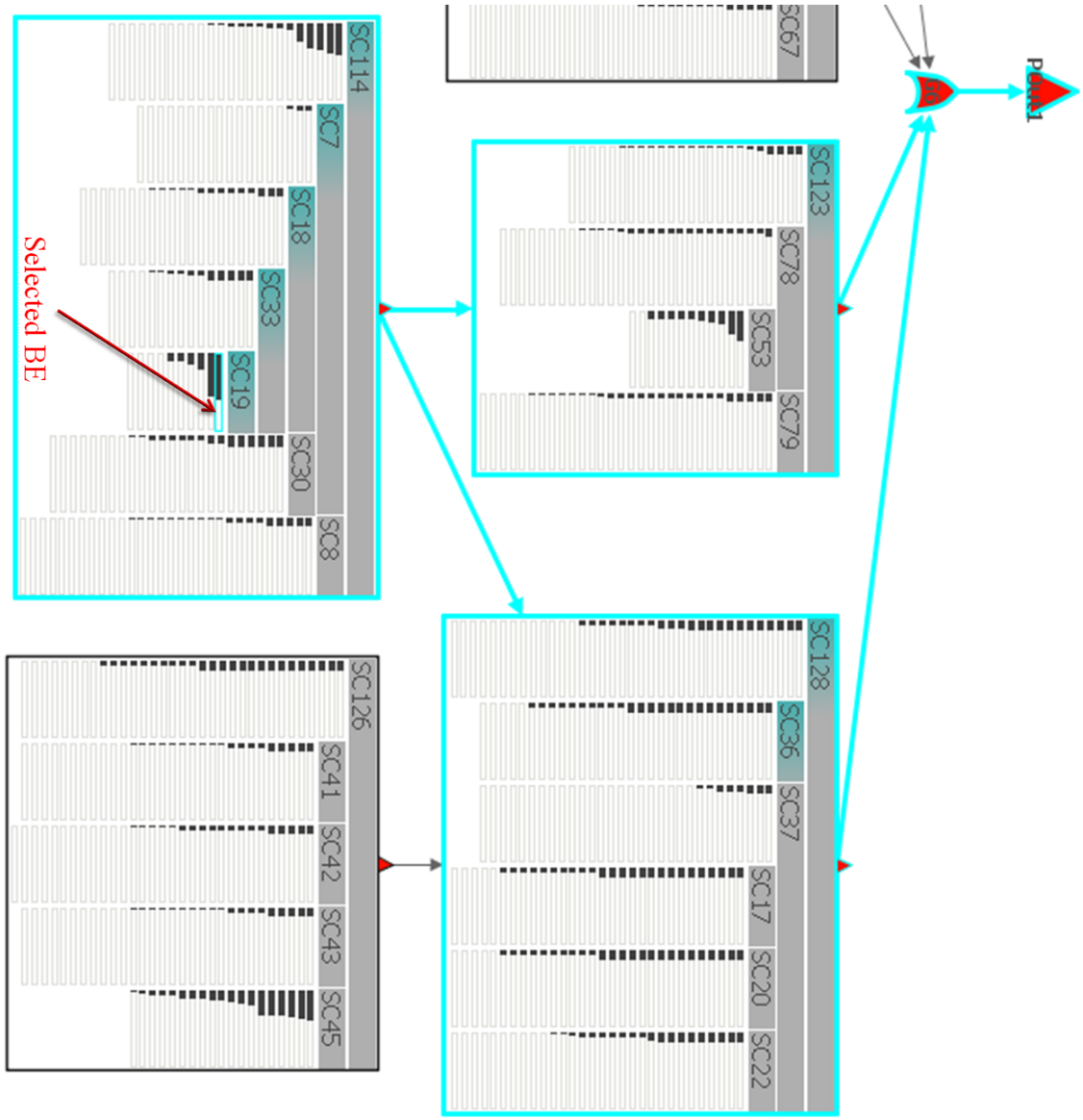


Figure 6.24: Application example 3. A partially enlarged influence of the important basic event along logical structure.

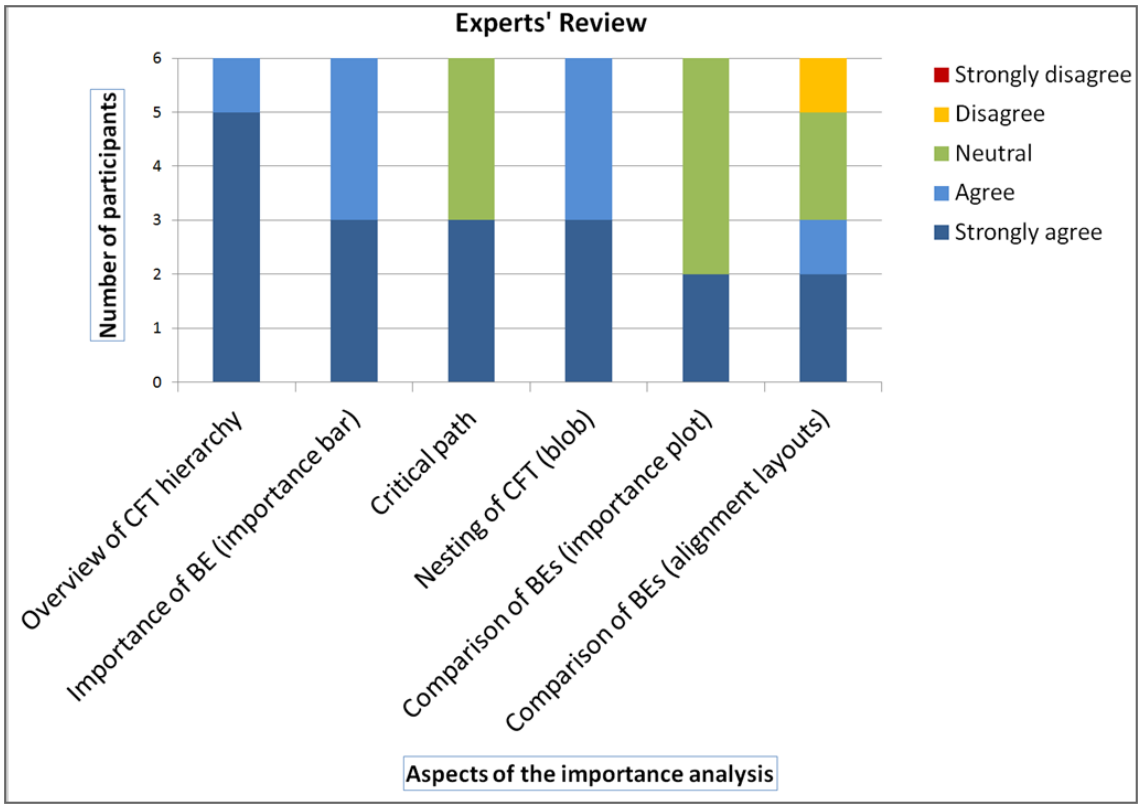


Figure 6.25: Experts' review.

influenced by another flow from the CFT component "SC114". Both components "SC123" and "SC128" output their failures to the top event of the system-level component by passing gate "G6".

This example illustrates how to explorer the information of the importance of basic events and the failure propagation when focusing on the logical structure of the system-level CFT component. In contrast, using the ordinary representation concept engineers have to frequently switch views when estimating basic events in the CFT structure, as well as invest a lot of effort into analyzing the deeper-nested subcomponents.

6.3 Evaluation

In order to estimate our visualization, an expert review was carried out. There were six participants with degrees in computer science that worked on the CFT analysis at the university. We introduced our visualization approach and then they were allowed to go ahead and experience using it. We provided tasks based on the importance analysis of the user experience. Finally, we asked the participants to fill out a Likert-scale questionnaire.

Basically, the participants gave positive feedback on our visualization (Figure 6.25). The participants commented that the critical CFT components can be easily identified with the help of the architectural view. There were also doubts regarding the iceray diagram. A participant suggested to use the sector diagram instead of the

iceray diagram for reducing the space requirement. Some participants pointed out that the node-link layout of the logical structure was too sparse and takes up too much space. The importance bar received positive comments, due to its intuition and the universal comparability. The participants commented that the quantitative failure flow along the CFT structures can be intuitively analyzed by the color of CFT nodes.

The participants considered that analysis for multiple critical paths was performed well by the highlighting of critical paths. Three participants suggested to use rainbow colors instead of blue and light blue for multi-level critical paths. The component expansion concepts were also strongly preferred because the investigation of continuous failure flow was possible and the analysis for the deeper-nested sub-components did not need to go through all parent components. The concept of aggregation blobs was regarded as being effective for identify the range of components. The participants also commented that the dashed curve linking the CFT structure to the architectural view was not suitable because it confused the referencing lines with the input lines. The participants thought that the importance plot was a effective auxiliary view for determining multiple important basic events. The basic event alignment concepts were also positively evaluated because of the intuitive layout for comparison. But the participants also pointed out that both layouts, particularly the global layout, required a significant amount of screen space and may not be suitable for large scaling.

Chapter 7

Visualization of Safety Improvement Process

7.1 Visual Support for Safety Improvements

The safety improvement process works on the construction of solutions in order to reduce the failure probability of the top event to an acceptable value. Each solution consists of a group of system design modifications. According to the design concept discussed in section 4.3, we propose a visualization approach associating a core risk-reduction plot with a set of linking views in order to visually support the safety improvement process. In this section, the implemented visualization is introduced with respect to the application scenarios of the safety analysis.

7.1.1 Representing Improvement Solutions

The core of the visualization approach is the risk-reduction plot combining a decision tree, a scatter plot, and other graphical properties (Figure 7.2 (2)). The risk-reduction plot is visually associated with the CFT structure for the meaningful context information. We associate the CFT view of the visualization of the importance analysis (VisQSA) introduced in section 6.1 with the risk-reduction plot. This allows to link information between the risk-reduction plot and the CFT view. Basic events are projected horizontally along the x-axis of the risk-reduction plot according to their locations in the CFT view. The achieved change in risk in terms of failure probability is projected along the y-axis.

To conveniently identify the important basic events (referring to the data R1 introduced in section 4.3), bars are presented along the x-axis on top of the plot as indicators of basic events (the more important a basic event, the longer the bar) (Figure 7.4). Labels are printed on the indicators. In case the basic events are hidden in the unexpanded CFT components in the CFT view, we only present the labels of the unexpanded components rather than those of the invisible basic events. The bar for an unexpanded component depends on the maximal importance of the basic events included by the component. More interactive associations between the risk-reduction plot and the CFT view are introduced in section 7.1.2.2.

The important basic events correspond to the vulnerable (physical) parts of the

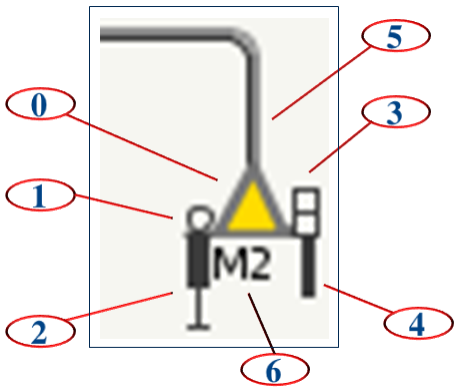


Figure 7.1: Risk-state node. (0) Risk state of a system with respect to the failure probability of the top event. Color indicates the unreliability level of the top event. (1) Modification type. A circle indicates substitution concept; a small triangle indicates redundancy concept. (2) Modification value. (3) Modification cost. (4) Cost-effectiveness of the modification. (5) An edge connecting a node with its predecessor node. The vertical part represents the resulting reduction of failure probability of the top event. (6) Modification ID.

system. Based on domain knowledge, engineers may determine possible design modifications in order to address the vulnerabilities. In order to integrate the significant data of the individual modification (Table 4.8), we propose a risk-state node that comprises of a central triangle icon and four attached visual items (Figure 7.1) as follows:

- Triangle icon: the actual risk state of the system (referring to require data R5).

  This is the most significant data based on the updated CFT. The fill color of the triangle represents the unreliability level of the top event. Using colors, one may quickly estimate the criticality of the risk state of a specific modification. When the color becomes green, the risk reduction can be finished and the corresponding solution is completely constructed. This icon also represents the milestones of the risk reduction during the safety improvement process, e.g., by which step the failure probability is reduced from critical level to moderate level. Figure 7.2 (2) shows an example that regarding the highlighted solution "S5", the system safety may be reduced from the critical level (red) to the moderate level (yellow) after performing the third modification "M17" (i.e., the yellow node on the far left). The ID of the modification may be presented under the risk-reduction node (Figure 7.1 (6)).
- Item 1: modification type (referring to R2).

  The types are nominal data that can be effectively represented by the graphical properties of position, color, texture, connection, density, and shape. The graphical properties of position, color and connection have already been used in our visualization. Because the size of the triangle icon is small, the graphical property of density is not suitable, too. Thus, we use shapes to represent the basic types of design modifications: a circle represents a component substitution, whereas a small triangle represents the introduction of redundant

components.

- Item 2: modification value (referring to R3).

  This value represents a reduction between the failure probability of the original basic event and that of the updated basic event (or sub-tree). We have designed a bar graph in order to present this information in an intuitive way. The bottom line of the bar indicates the failure probability of the initial basic event. The filled part shows the new failure probability. If engineers replace the original hardware, the filled part represents the failure probability of the new part. If engineers apply the redundancy concept, filled part represents the failure probability of the new sub-tree constructed for the redundant parts. The bar reflects the reduction proportion of the failure probability. The item provides information about the context under which the modification has been applied. For example, following the substitution approach, it is possible to intuitively compare the existing part with the substitution part in terms of failure probability.

- Item 3: modification cost (referring to R4).

  We propose a scale bar to visualize the cost not only for the comparison of modifications, but also for the investigation the absolute cost value. Engineers are allowed to define the scale of the bar, i.e., the unit value of the block of the bar, e.g., each block represents 10 dollars. Engineers are allowed to use an alternatively logarithmic scale instead of the linear scale.

- Item 4: cost-effectiveness of the modification (referring to R7). A bar is introduced to represent the cost-effectiveness of the design modification. The more cost-effective the modification, the larger the bar. Engineers are allowed to adjust the parameters of the bar, e.g., the unit Length of the cost-effectiveness value, in order to adapt to different systems. Liner scale and logarithmic scale are applied to the bar.

Additionally, if a node is currently the last step of a solution, the ID of the solution is printed next to the node For example, in Figure 7.6 modification "M2" is currently the last step of solution "S1".

We design the risk-state node taking the following concepts into account. There are two possible ways to compose the central triangle icon and the visual items: the inside composition strategy and outside composition strategy. When placing the visual items inside the central icon, the icon needs to be enlarged. In this case, a large icon cannot precisely indicate its position (in the plot) that represents significant semantic meanings of the analysis process. Thus, we apply the outside composition strategy. We place the four visual items closely around the central icon. The visualization properties with respect to the method of a modification (items (1) and (2)) are placed in the left side; the factors for evaluation of the modification are represented in the right side (items (3) and (4)). This way, engineers may investigate the modification method and the evaluation of the modification in their respective sides.

We connect a new risk-state node with its direct predecessor using a two-part orthogonal edge (Figure 7.1). A horizontal line between the predecessor node and the horizontal position of the new risk-state node represents the subsequence of the modifications. The vertical part of that line represents the reduction of the failure

probability of the top event resulting from the corresponding modification (referring to R8). We use a rounded corner to connect both parts in order to address the edge crossing issues. When there are alternative modifications for a basic event, multiple risk-state nodes are allowed to be created. In this case, between these nodes, there are equal distances for addressing the overlapping issues of nodes as well as edges (Figure 7.7: "M2" and "M3").

We additionally provide a horizontal green line in the lower part of the risk-reduction plot in order to indicate the goal value of the overall safety improvement. This facilitates to assess how far the failure probability of the top event at a specific node from the goal value is. When the system risk becomes acceptable by a modification, the top event will have the failure probability that equals to the goal value. In this case, the risk-state node stops at the green horizontal line. In addition, when selecting a solution, its risk-state nodes and the connecting lines will be highlighted.

7.1.1.1 Identifying optimal Modifications

Engineers may identify the optimal ones from the alternative modifications in the construction process according to the criteria described in section 2.1.6.1. Either the visual item (4) of the risk-state node or the vertical part of edges can be used for identifying the optimal modifications. The non-optimal modifications are rejected and the corresponding branches are terminated by filling up the last node with black color. When reviewing these nodes, engineers may quickly understand that the corresponding modification ideas have been considered and treated as unsuitable (Figure 7.7). Thus, engineers no longer consider successors of the modification.

7.1.2 Reviewing CFT Adaption

7.1.2.1 Adapting the main CFT Structure.

A design modification leads to the adaption of the corresponding CFT that is represented by changing failure probabilities and/or the logical structure. Our system automatically updates the CFT model according to modifications in the background during the analysis process. In order to preserve the overview of the vulnerable basic events involved in a solution, the initial logic structure of the CFT view is maintained during the safety improvement process. This helps to avoid disturbances caused by subsequently updating the CFT. When applying a modification, the color of the initial basic event node is adapted to the failure probability of the substitutional part or to the new constructed sub-tree for redundant parts. To show the influences of modifications along the CFT structure, the colors and the size of the corresponding nodes along the critical path are adapted. When selecting a risk-state node in the plot, the corresponding basic event node in the CFT structure is dynamically highlighted using a thick border.

7.1.2.2 Adapting CFT Components

When a CFT is large, the compact views (i.e., architectural views and simple rectangular symbols) are more space-efficient than the logical structures. We provide a

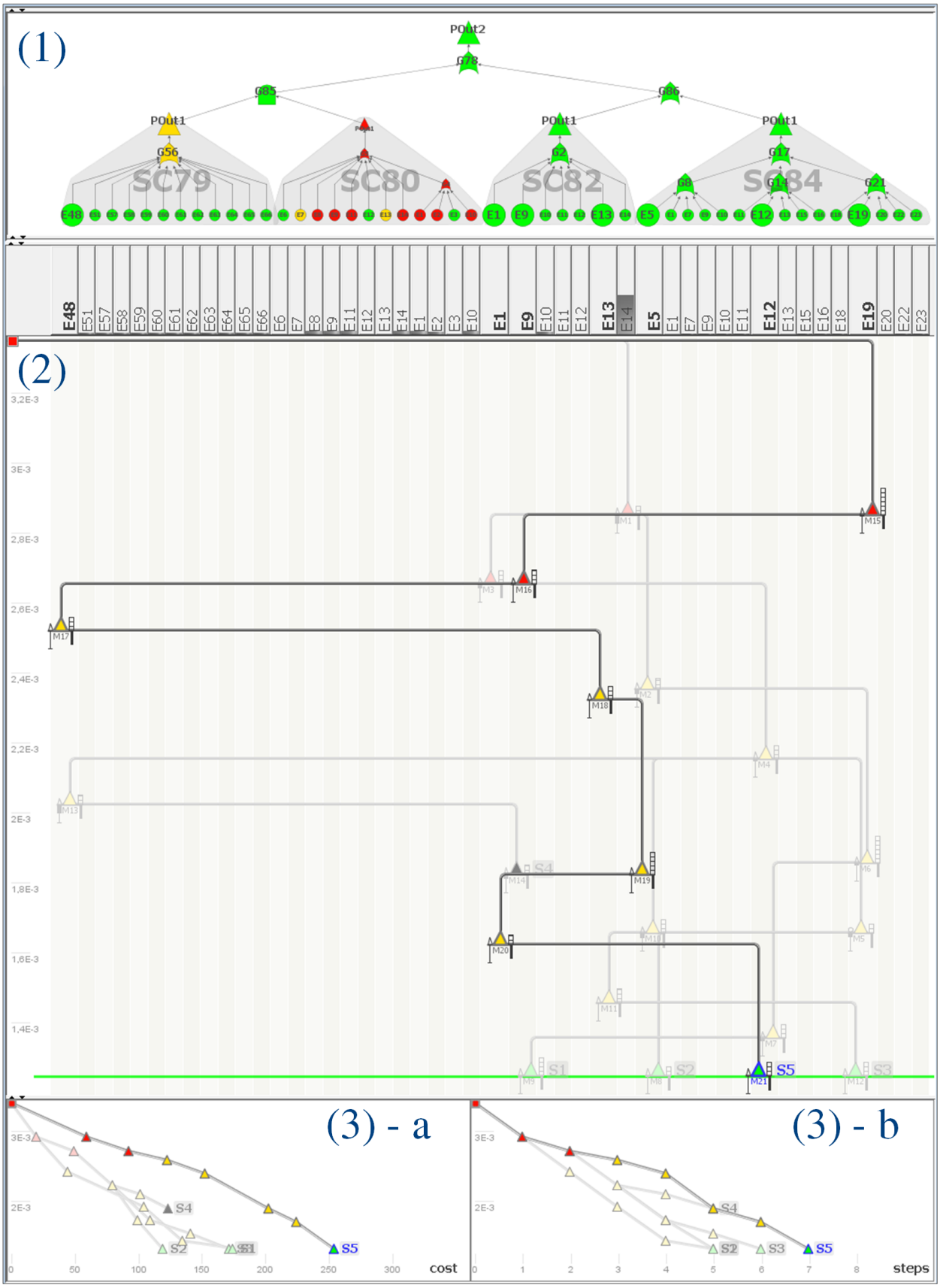


Figure 7.2: Visualization of the safety improvement process. (1) The associated CFT. (2) The risk-reduction plot. (3) The solution overview plots. This field consists of two alternative plots where the x-axis represents (a) the cost of solutions; (b) the steps of solutions.

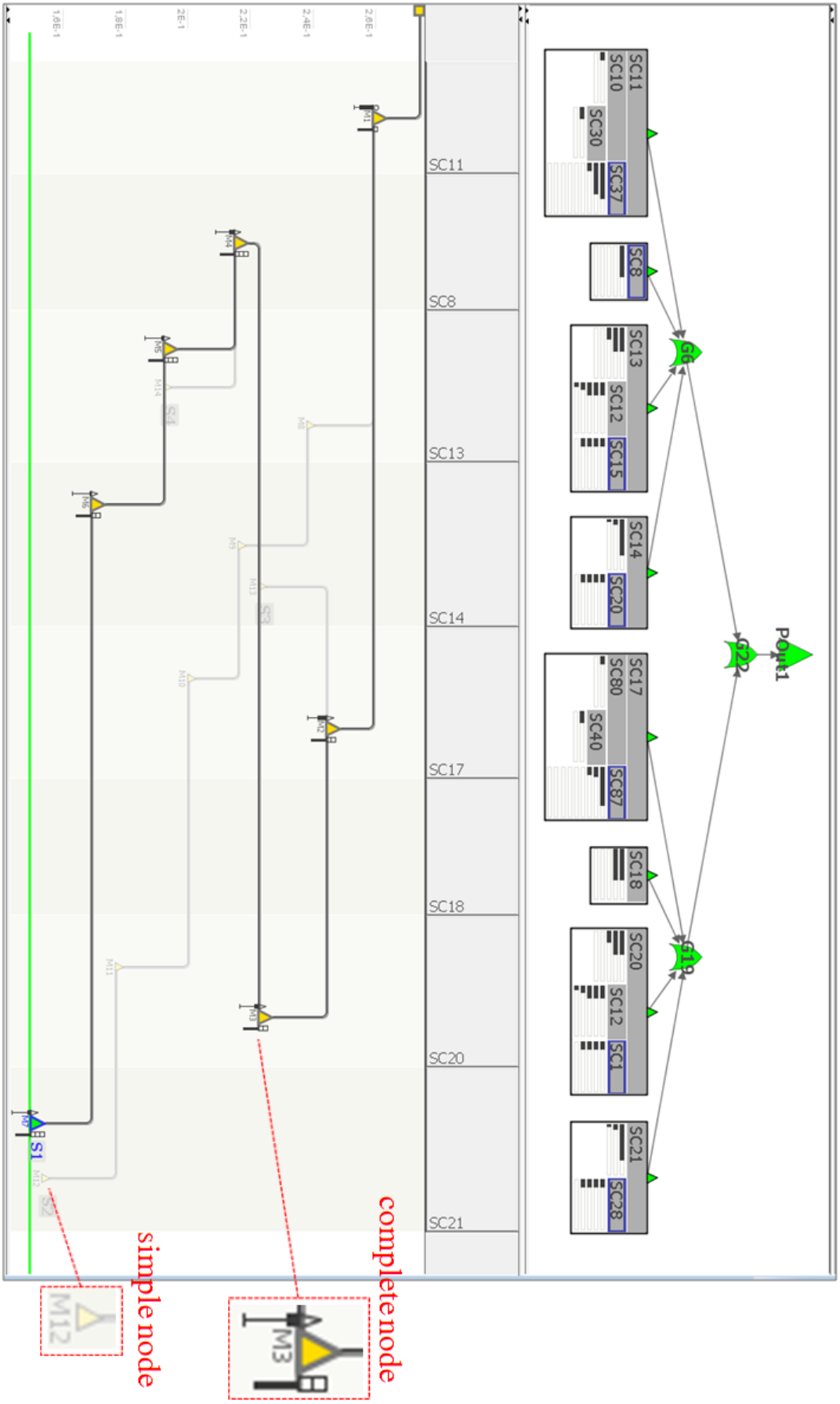


Figure 7.3: Adaption to the architectural view.

method to adapt the risk-reduction plot to the compact views. A CFT component represented using a compact view has an indicator on the top of the plot. The basic events of this component are aggregated under the indicator. In this case, in order to space efficiently display the nodes, the complete risk-state node is replaced with a simple triangle icon without additional attached visual items (Figure 7.3). Only when the node is focused on, the complete risk-state node will appear.

A solution is related to a group of basic events. In order to indicate the CFT components that directly include these basic events in the architectural view, the rectangles representing these components are marked with blue borders. Figure 7.3 shows an example. Solution "S1" is selected. The basic events relevant to this solution are distributed in the CFT components that are represented as rectangles with blue borders.

7.1.2.3 Reviewing the Change of the CFT Structure

We provide a details-on-demand pop-up view to represent the updated logical structure regarding the specific risk-state node (Figure 7.11). The pop-up view shows the structure of the CFT component that has the basic event corresponding to the currently selected risk-state node. The part related to the current modification is enclosed with a dashed border. This is particularly useful to intuitively review the design modifications that utilize the concept of redundancy.

7.1.3 Analyzing Improvement Solutions

We provide two linked plots to present the overview of the solutions with respect to the criteria of optimization of solutions (Figure 7.2 (3)). The y-axes of these plots represent that failure probability of the top event. The x-axis of the plots respectively represents the following data:

- cost of solution: for the criterion of the maximal cost-effectiveness. Because any complete solution reduces the failure probability of the top event to the same goal value, i.e., the effectivenesses are same, engineers may compare the total cost instead of the total cost-effectiveness for identifying the optimal solution.
- steps of solution: for the criterion of the minimal modification steps.

Each modification has a simple triangle node on the linked plots. Engineers may analyze patterns of the risk reduction with respect to cost and the number of modifications in the linked plots. Colors of nodes are same as those in the risk-reduction plot. A brushing-and-linking interaction is provided between the risk-reduction plot and the linked plots in order to analyze solutions from different points of view.

7.2 Application Scenarios

There are application examples intended to illustrate the use of our visualization approach with respect to two important aspects: construction of solutions and the analysis of existing solutions.

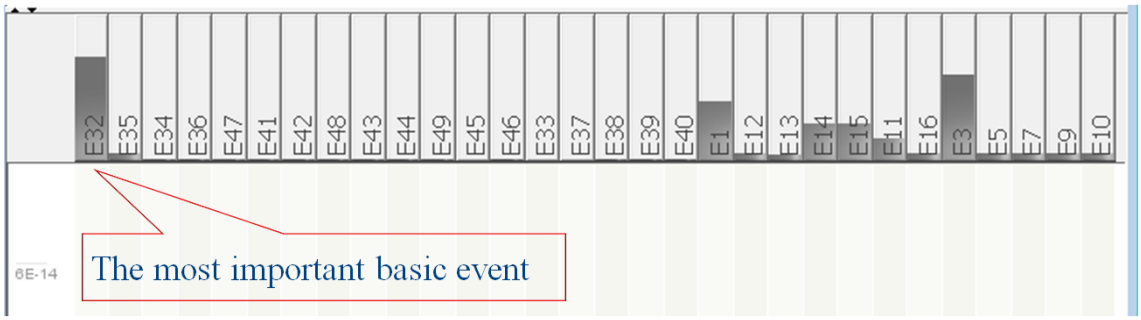


Figure 7.4: Application example 1. The initial state. The importance of basic events are estimated by analyzing the bars on the top of the risk-reduction plot. The basic event "E32" is the most important because its bar is larger than other basic events'.

7.2.1 Example 1

This example presents a scenario for constructing the improvement solutions. The goal is to identify the most cost-effective solution(s).

7.2.1.1 Dataset and Configuration

The applied data originates from a CFT of a safety-critical sub-system of the robot RAVON as an example by the project ViERforES [201]. This CFT contains 30 basic events and 4 CFT components. This data are used in Example 1 and Example 2. The initial failure probability of the top event amounts "6.7e-14", and the specified goal value is "1.5e-14". We apply the linear scale to the y-axis of the risk-reduction plot in the example. For the risk-state node, we apply the logarithmic scale to the bar representing cost-effectiveness and the bar representing the modification value because the bars for small values are difficult to read when using linear scale. We set the unit value of the cost bar (of the risk-state node) to "10", i.e., each block of the bar represents 10 cost units. The unreliability levels are defined as follows:

- critical level (red): (2e-5, 1]
- moderate level (yellow): (1.5e-14, 2e-5]
- acceptable level (green): (0, 1.5e-14]

7.2.1.2 Analysis Process

There are following main steps in the improvement process for reducing the risk of the system to the goal value.

1. In the first iteration, by the initial system, the basic event "E32" is treated as the important one by examining the bars on the top of the risk-reduction plot (Figure 7.4). Based on the domain knowledge, we decide to replace the identified vulnerable physical part having a failure probability of "4e-4" with one having a value of "2.5e-4". The cost of this modification amounts to 20 units. The CFT is automatically updated according to the modification. As a result, a risk-state node for modification "M1" appears on the risk-reduction plot (Figure 7.5). The vertical position of the node shows that the overall

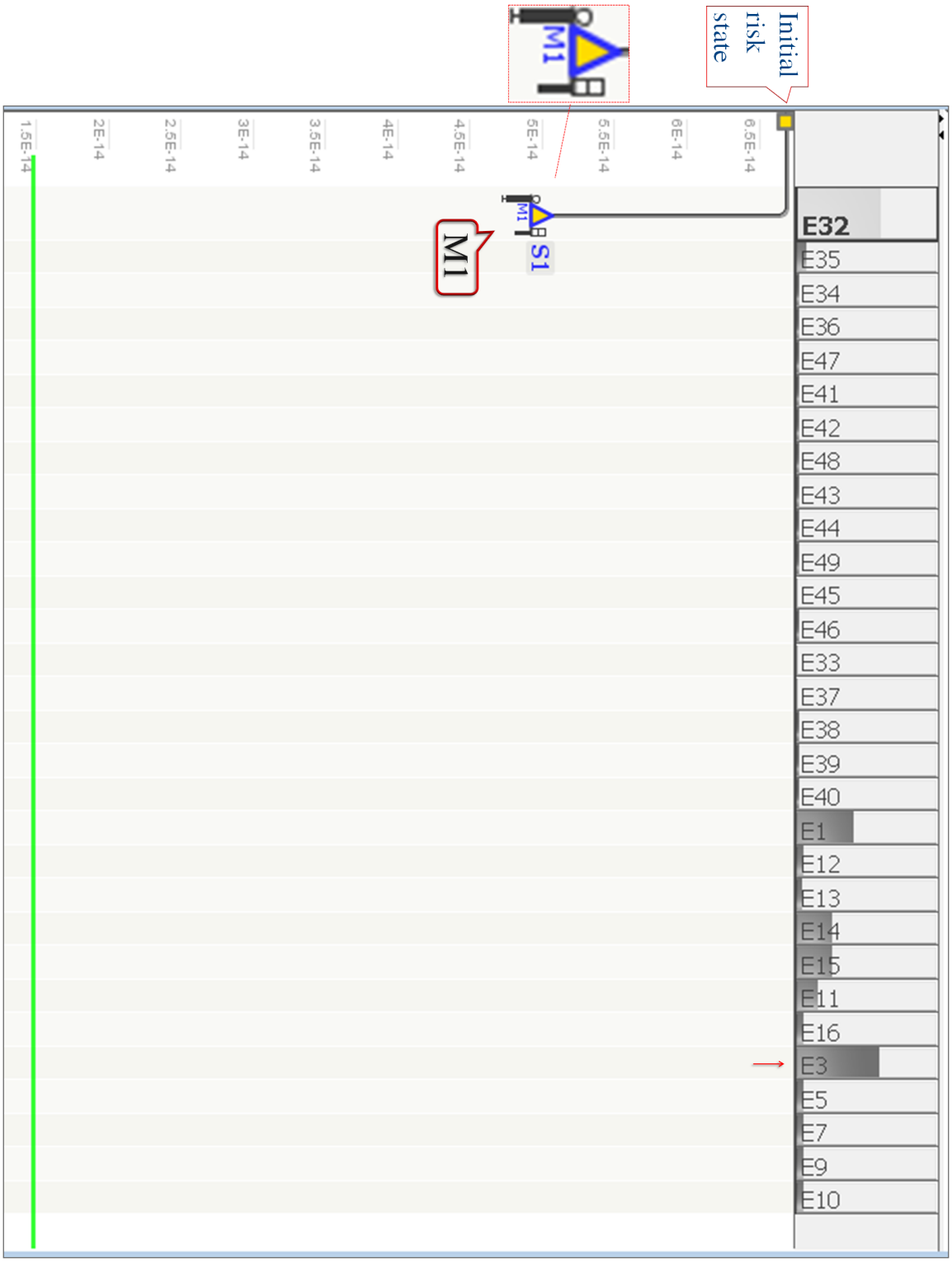


Figure 7.5: Application example 1. The first iteration. The risk-state node of modification "M1" represents the first design modification that corresponds to the basic event "E32".

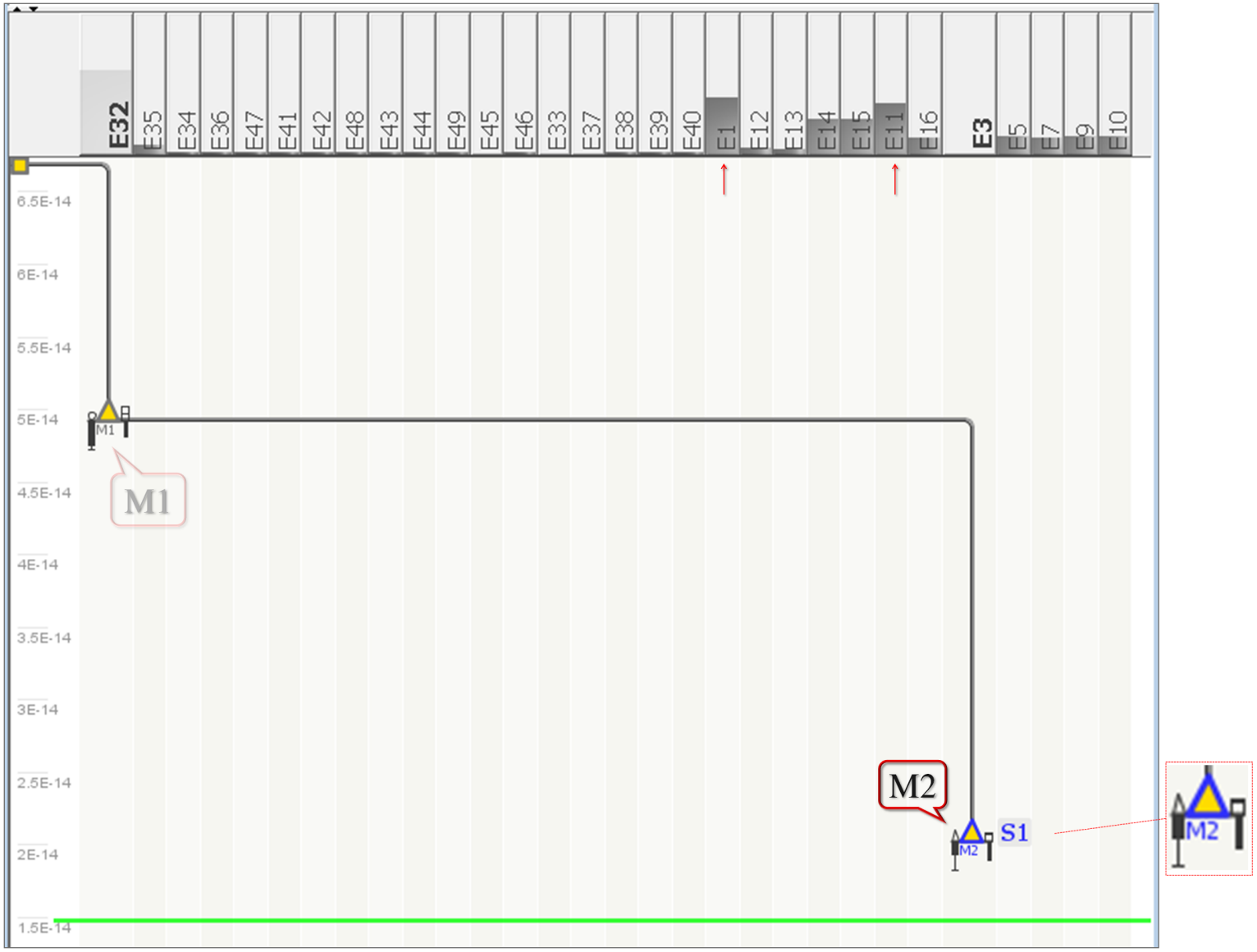


Figure 7.6: Application example 1. The second iteration. The risk-state node of modification “M2” is generated for applying the redundancy concept to the basic event “E3”.

Figure 7.7: Application example 1. The second iteration. The risk-state node of the modification "M3" is generated for applying the substitution concept to the basic event "E3". By comparing the cost-effectiveness of both modifications, we decide to apply the modification using redundancy concept.

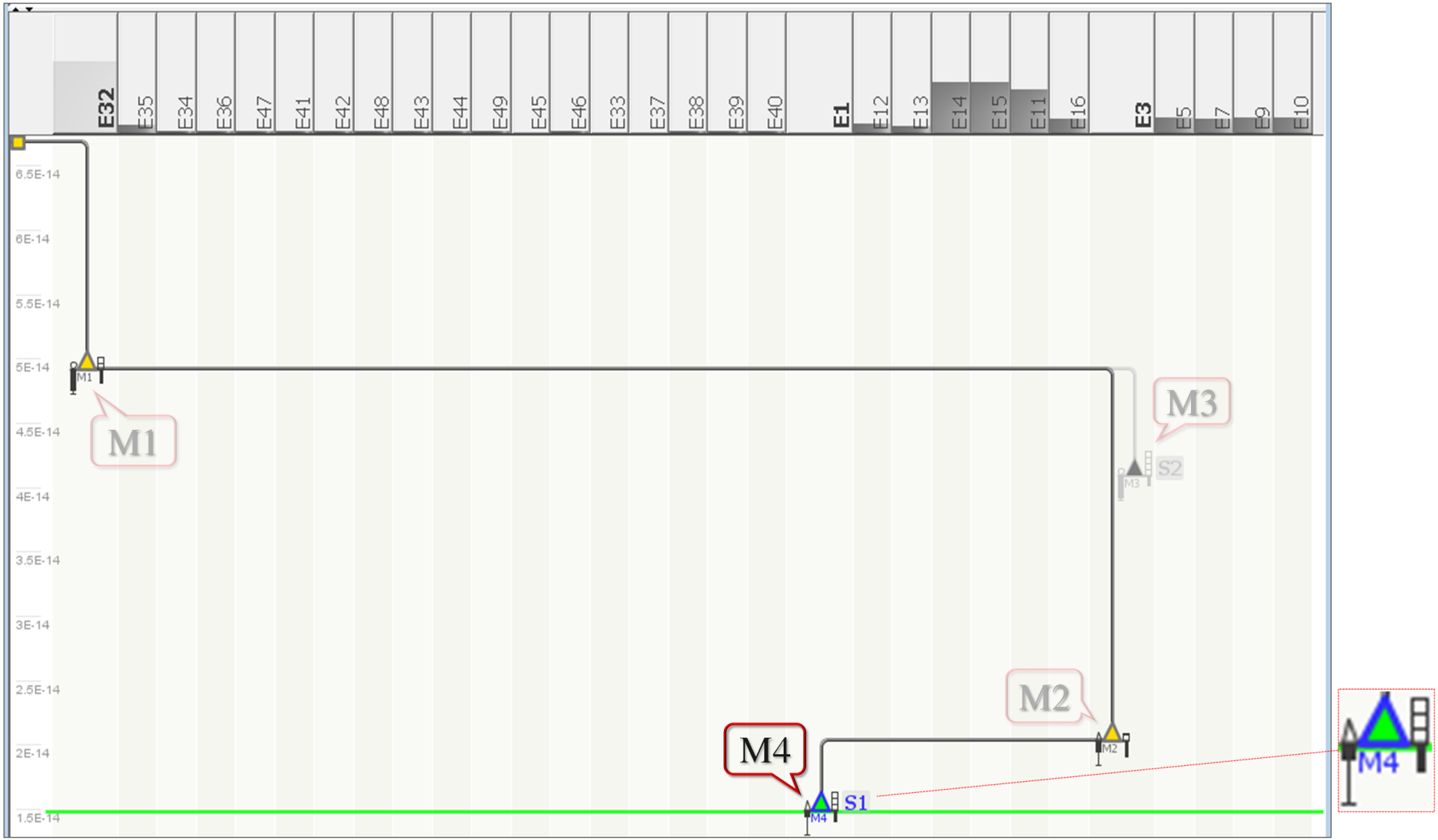


Figure 7.8: Application example 1. The third iteration. The risk-state node of modification "M4" is generated for applying the redundancy concept to the basic event "E1".

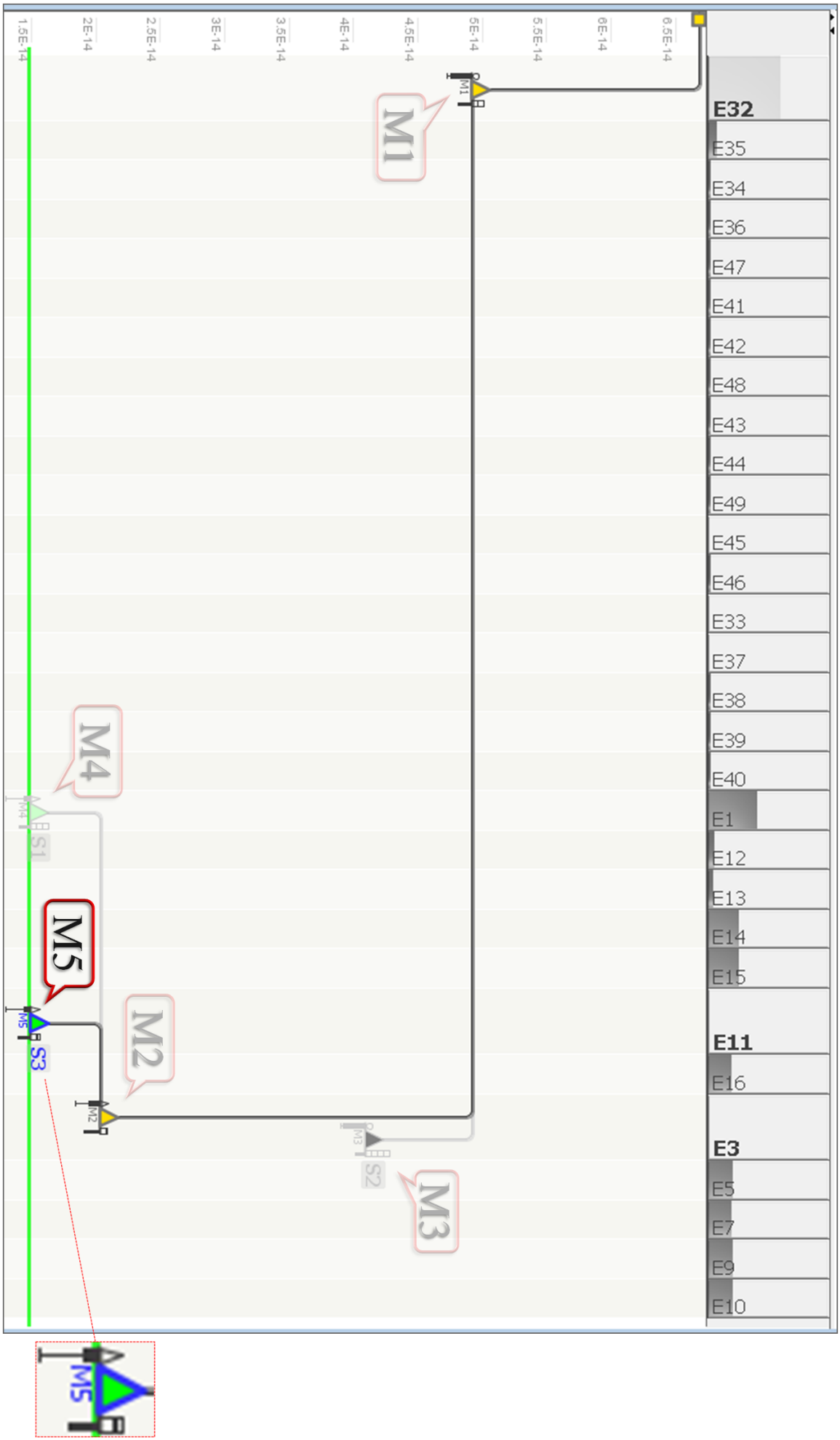


Figure 7.9: Application example 1. The third iteration. The node of modification "M5" is generated for applying the redundancy concept to the basic event "E11".

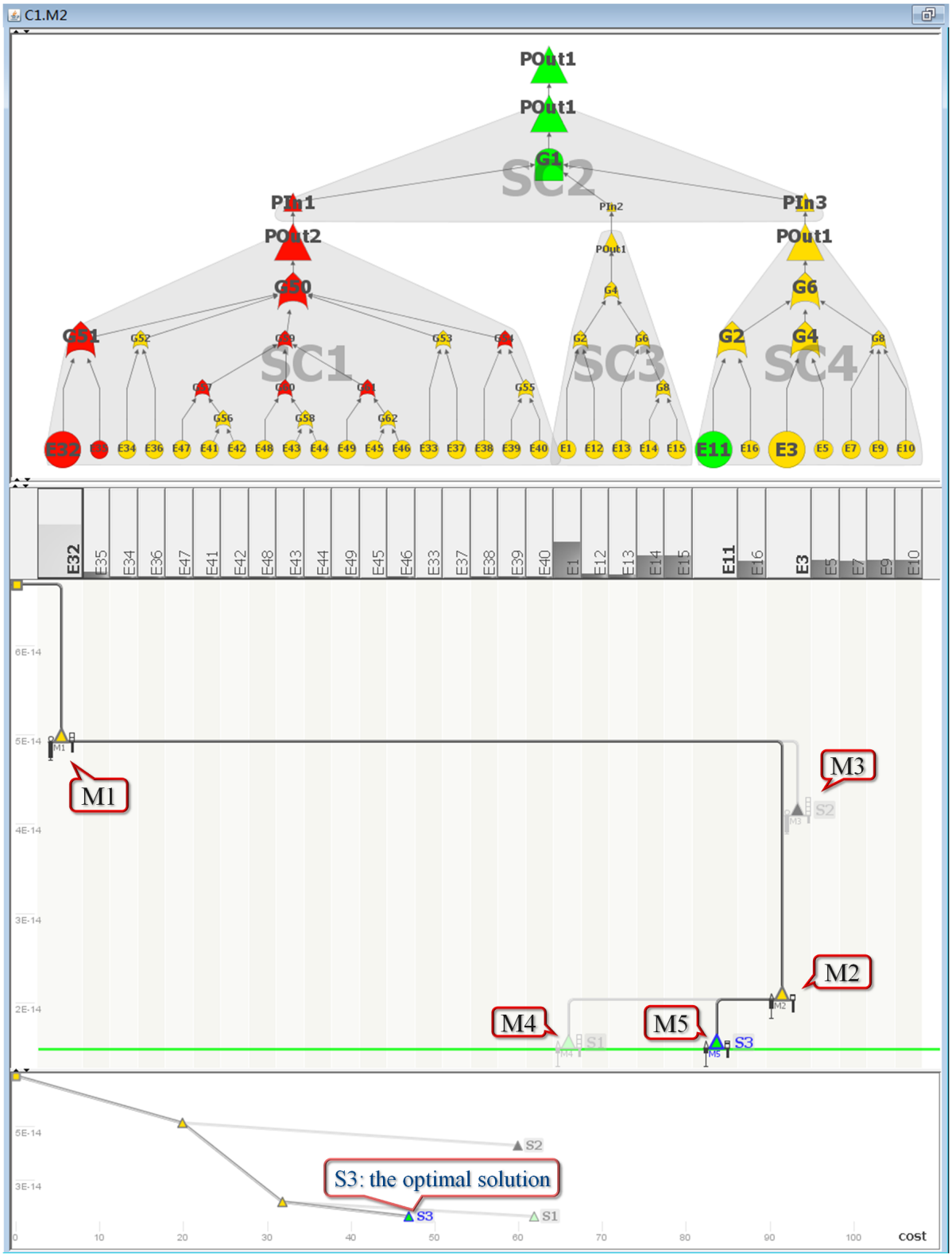


Figure 7.10: Application example 1. Overview of the results of the risk-reduction process.

failure probability is reduced to "4.9e-14". Solution "S1" containing "M1" is being constructed because "M1" leads to a new branch. The ID of "S1" appears next to the node. The risk is not acceptable yet because the color of the node is not green. Then we need to perform the next round.

2. In the second iteration, the important basic event "E3" is identified (Figure 7.5, pointed by a red arrow). There are two possible design modifications to address the basic event: one is to add an homogeneous redundant component (Figure 7.6: "M2") that reduces the failure probability of the top event to "2.03e-14" and causes the costs of 12 units; another one is to use a substitute (Figure 7.7: "M3") that reduce the failure probability of the top event to "4.1e-14" and causes the costs of 40 units. The cost-effectiveness bars show that "M2" is more cost-effective than "M3". Thus, "M3" is rejected and the corresponding branch is terminated. After this iteration, the failure probability of the updated CFT is not acceptable yet.
3. In the third iteration, two important basic events "E1" and "E11" are identified (Figure 7.6, pointed by red arrows). Although the basic event "E32" has a high importance, we do not consider it because this basic event has been previously addressed. The color of the bar on the top of the plot reflects this information: bar of "E32" is light gray (the practicable basic events have dark-gray bars). Based on the domain knowledge, we identify modification "M4" for "E1" causing the cost of 30 units (Figure 7.8) and modification "M5" for "E11" causing the cost of 15 units(Figure 7.9). By either modification the system risk can achieve the required goal value of the top event.
4. As a result, there are two available solutions constructed in the safety improvement process with respect to the cost-effectiveness (Figure 7.10) (solution "S2" has been rejected). We still need to determine the optimal solution. We compare the total cost of the solutions. The fact that the cost of solution "S3" is less than the cost of solution "S1". It concludes that solution "S3" is the most cost-effective way to improve the system safety.

This example illustrates the complete procedure of identifying the optimal improvement solution using our visualization. Additionally, in Figure 7.10, the labels of the CFT components printed in the blobs show that the basic events related to solution "S3" are distributed over two CFT components: "SC1" and "SC4". The system components corresponding to these CFT components need to be focused on in order to improve the system design with respect to the safety.

7.2.2 Example 2

This example presents an application that engineers review the modifications of solution "S3" constructed in Example 1. The objective is to investigate the design modification that introduces the largest impact on the top event.

7.2.2.1 Analysis Process

The objective is achieved by the following steps:

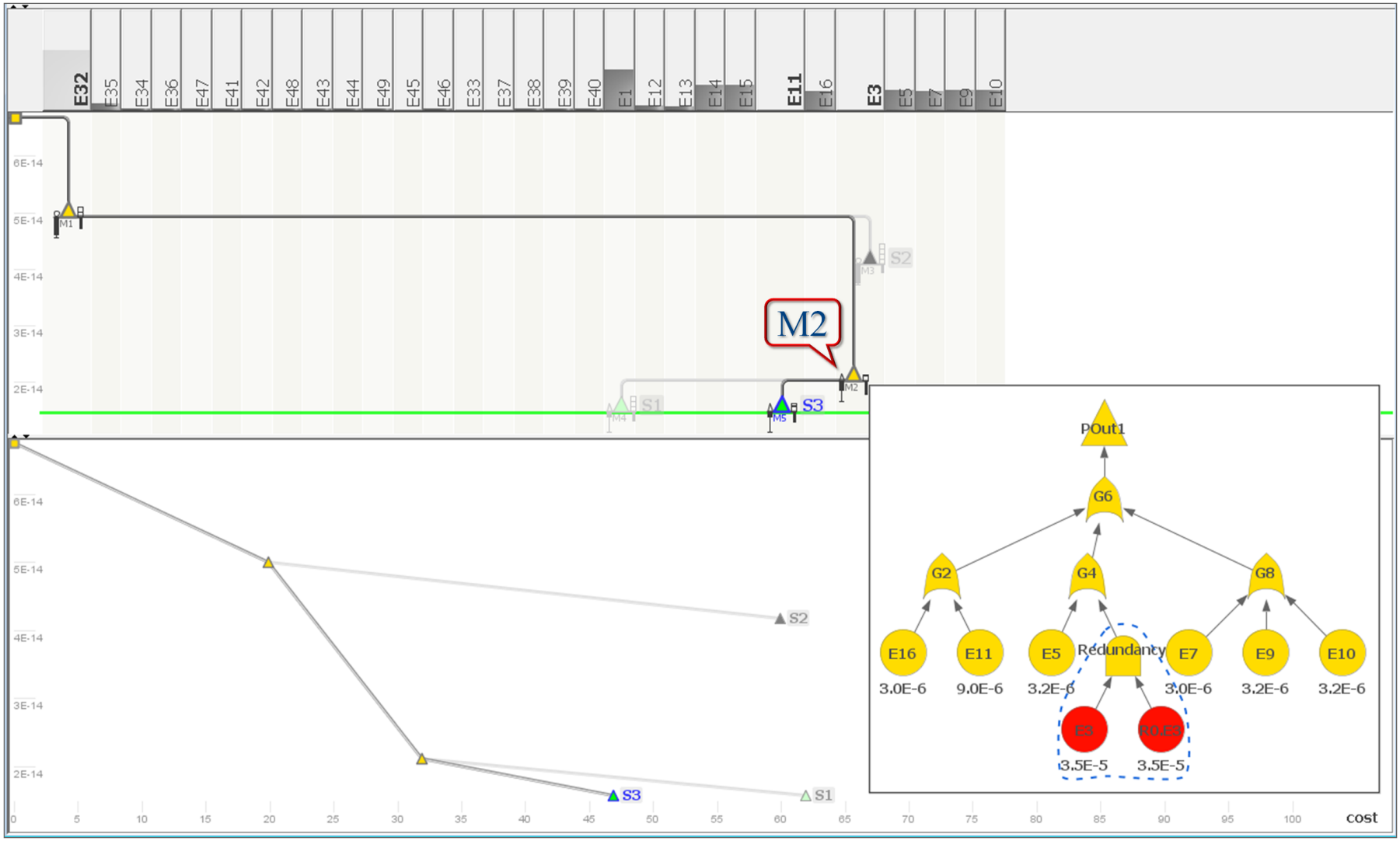


Figure 7.11: Application example 2. The pop-up window shows the detailed structure of the updated CFT component that associates with modification "M2".

1. We identify that modification "M2" causes the largest reduction of the top event (Figure 7.11) by directly comparing the vertical line of the edges linking to "M1", "M2", and "M5".
2. The symbol representing the modification type (the small triangle on the left upper of the risk-state node) explains that the modification applies a redundancy method.
3. By clicking on the node, the pop-up windows is shown for representing the updated structure of the CFT component "M2" (Figure 7.11). A sub-structure inside a polygon with a dashed border represents the redundancy relations of basic events. The sub-structure has two basic events: the original basic event "E3" and the new basic event corresponding to the added redundant component. An AND-gate connects both basic events to build a parallel redundancy.

This example shows how to analyze the existing modifications.

7.2.3 Example 3

Reviewing the constructed improvement solutions with respect to the CFT structure may support the understanding of design modifications for the future analysis or optimization. This example demonstrates an analysis task that is to identify the CFT components related to the optimal solution having the minimal cost. The pattern with respect to the distribution of design modifications belonging to the optimal solution over the CFT components is also of interest in the example. In addition, we also try to analyze the influence of the specific modifications along the logical structure of CFT components.

7.2.3.1 Dataset and Configuration

In this example, we apply a CFT model based on the robot RAVON that has 145 basic events. There are 8 previously constructed improvement solutions. The initial failure probability of the top event amounts "0.014", and the goal value is "1e-2". The lower and upper bounds of the importance bar are respectively set to "4e-4" and "1". We use the logarithmic scale for the importance bars. The unreliability levels are defined as follows:

- critical level (red): (0.5, 1]
- moderate level (yellow): (1e-2, 0.5]
- acceptable level (green): (0, 1e-2]

7.2.3.2 Analysis Process

The tasks are completed by performing the following steps:

1. We first identify the optimal solution. By viewing the cost plot, solution "S1" having the minimal cost is identified (Figure 7.12 (2)). Although solution "S6" (on the left of "S1") has the an equally low cost, this has been rejected and not a available solution. We learn this from the black color of the triangular node.

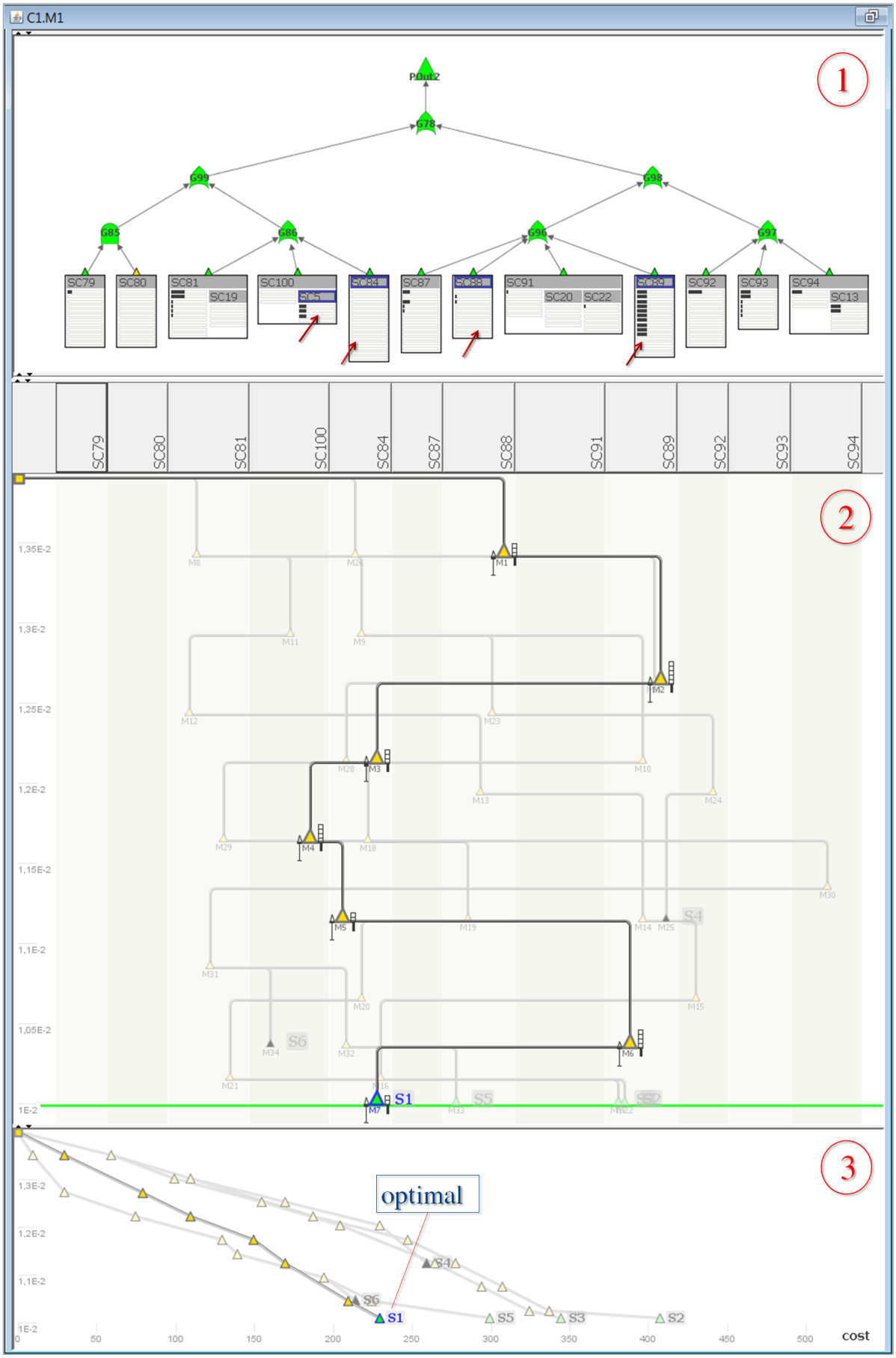


Figure 7.12: Application example 3. The CFT view shows the primary components. A partially enlarged view is shown in Figure 7.13.

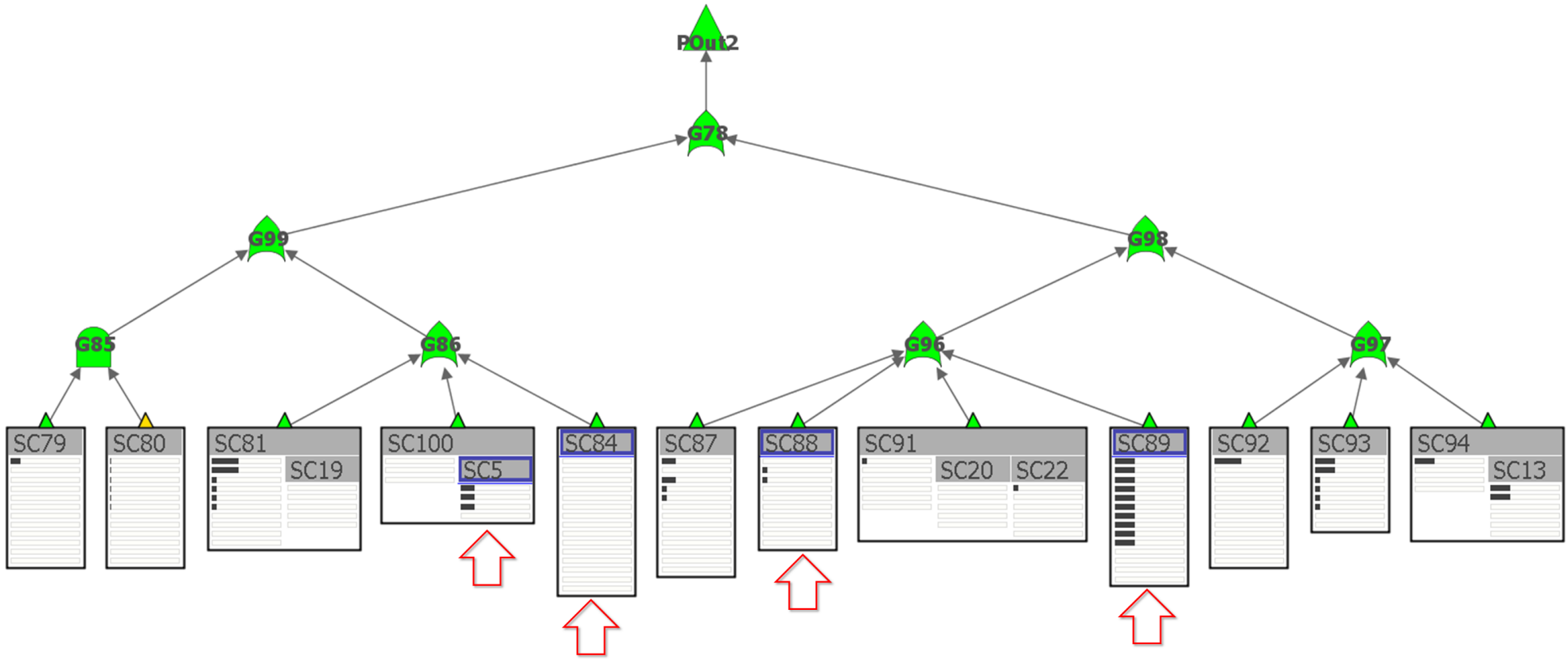


Figure 7.13: Application example 3. A partially enlarged view of the structures of the CFT components in Figure 7.12. The rectangles with blue a border represent the CFT components that are involved by the currently analyzed solution.

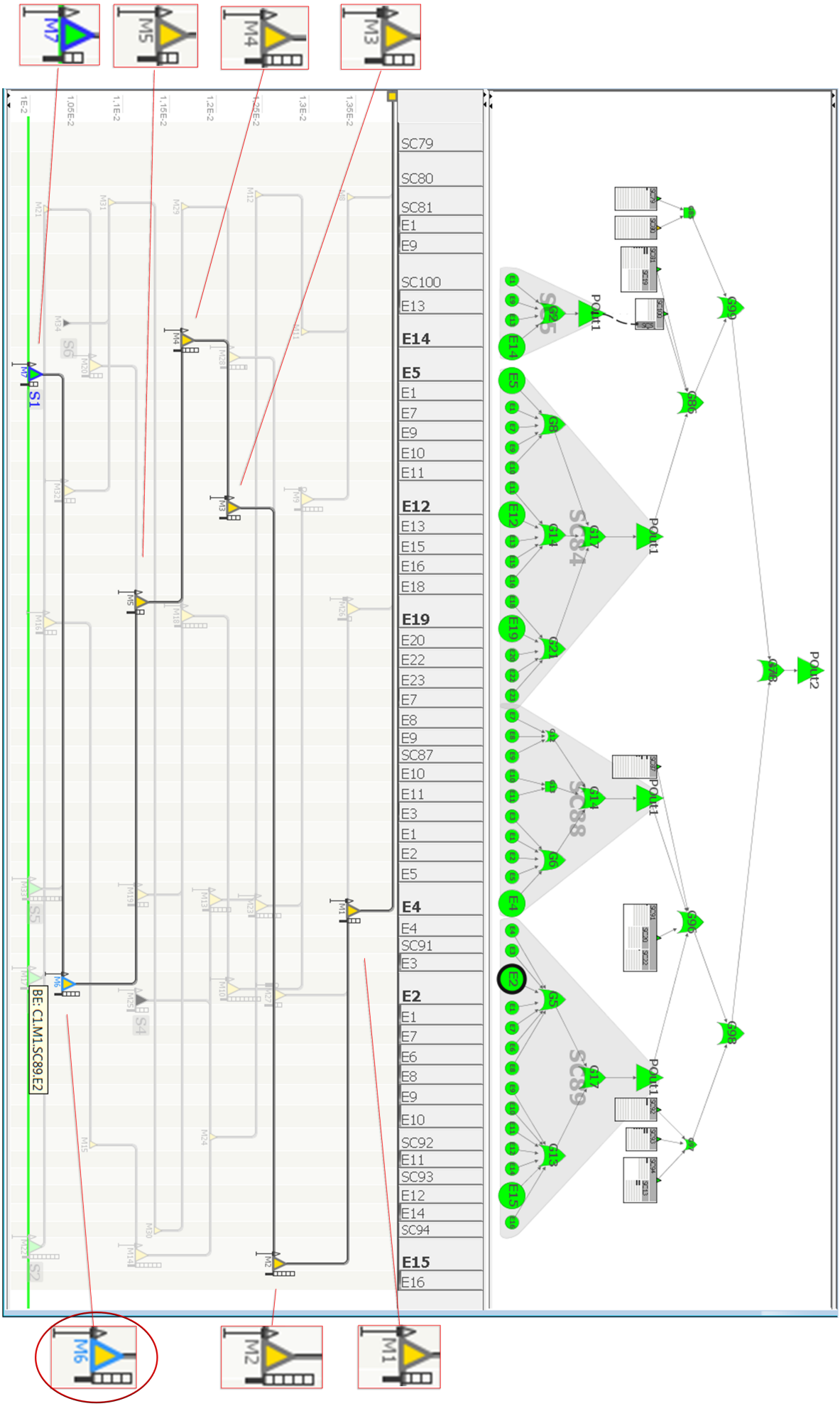


Figure 7.14: Application example 3. The logical structures of the CFT components that are related to the specific solution are displayed. A partially enlarged view is shown in Figure 7.15.

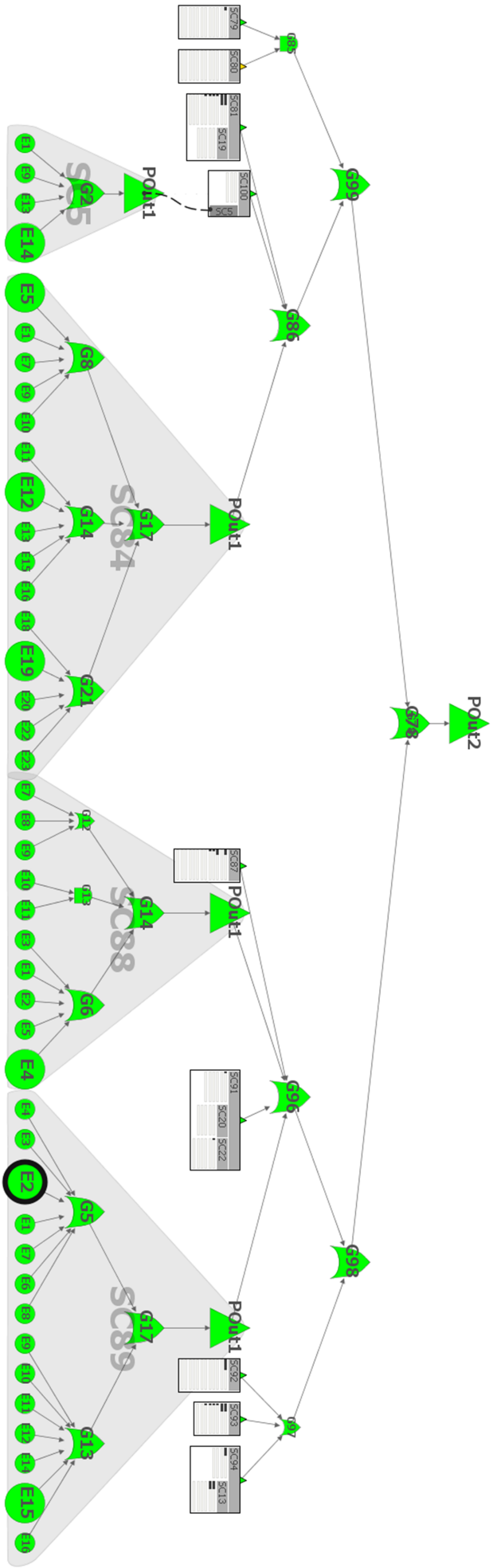


Figure 7.15: Application example 3. A partially enlarged view of the logical structures of the CFT components in Figure 7.14.

2. We then identify the CFT components related to the optimal solution "S1". For this step, we show the primary components using architecture views in order to efficiently use the screen space (in area 1). The CFT components involved in solution "S1" are indicated by blue borders in the architectural views. The resulting components are "SC5" (nested in "SC100"), "SC84", "SC88", and "SC89".

3. Then we investigate the detailed distribution of the modifications included by the optimal solution over the CFT components. There are two ways to identify the distribution: viewing the corresponding relations between the risk-state nodes and the architectural views in step 2 (Figure 7.12), or investigating the enlarged basic event nodes in the CFT view where gray blobs indicate CFT components. The first way provides a space-efficient view for the identification and the second way is more effective for the complex cases. For the second way, we need to show the logical structures of those four CFT components identified in last step. The visible logical structures show that the enlarged basic event nodes "E14", "E5", "E12", "E19", "E2", and "E15" are related to the design modifications belonging to solution "S1" (Figure 7.14). The resulting distribution is:

   - modification "M1" is related to the CFT component "SC88".
   - modifications "M2" and "M6" are related to the CFT component "SC89".
   - modification "M4" is related to the CFT component "SC5".
   - modifications "M3", "M5", and "M7" are related to the CFT component "SC84".

   According to the distribution, we may future perform the modifications directed towards the related components rather than sequentially modify the system design, if the related basic events are stochastically independent.

4. We finally investigate the failure flow of the basic events. This may provide a meaningful context for analyzing the effect of the design modifications with respect to the logical failure flow. By selecting any risk-state node, we may identify the corresponding basic event node in the CFT view, which is indicated by a thick border (Figure 7.14). We analyze modification "M6" that is related to the basic event "E2" as an example.

   With the help of the label printed on the blob, we identify that the corresponding CFT component represents "Actuator" (Figure 7.16). The basic event "E2" represents the failure "Steering Wheels are defective". The basic event may cause the intermediate failure at gate "G5", which represents "Front Steering Engine is defective". The failure at gate "G17" below the out-port of the component represents the failure "Actuator works incorrectly".

   In addition, we notice that the modification method is the application of the redundant parts because the icon representing the type of modification "M6" is a small triangle (Figure 7.14). The design modification "M6" may be understood as follows: it needs to add identical steering wheels to the front steering engine of the actuator in order to improve the safety of the mobile robot.

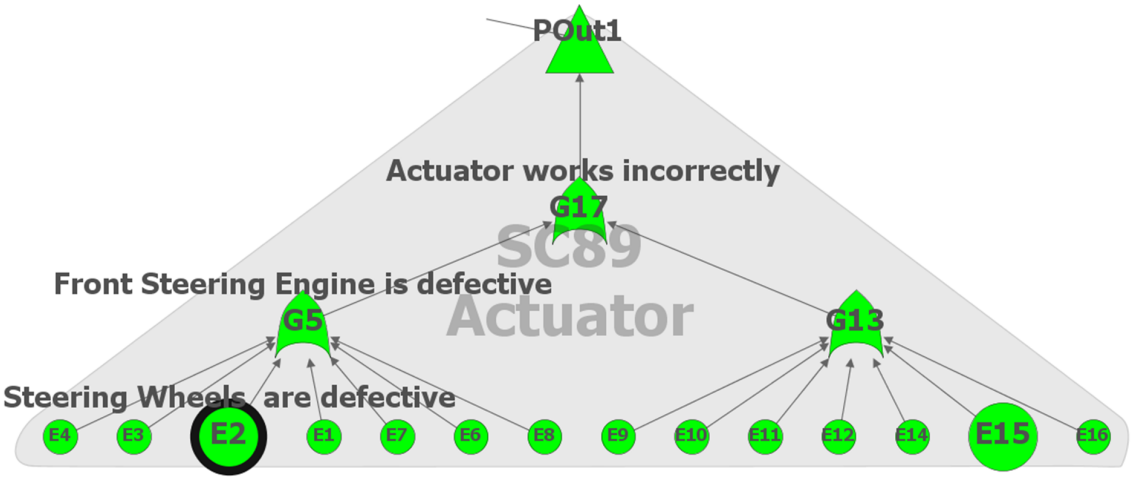


Figure 7.16: A partially enlarged logical structure of the CFT component "SC89". The failure flow of the basic event related to modification "M6".

This example presents that the association between the risk-reduction plot and the CFT structure facilitates to obtain the overview of the design modifications of the specific solution. In this way, engineers may also analyze the related critical paths of the basic events involved in the solution.

7.3 Evaluation

We have performed an expert review for evaluating our visualization approach. We invited four experts of the safety domain from the University of Kaiserslautern, all having profound proficiencies in the field of (component) fault tree analysis. We first introduced our approach to the participants, and then they were allowed to personally try the visualization functionalities. Tasks with respect to the safety improvement process were provided for the participants. Finally, the participants filled out a Likert-scale questionnaire for a qualitative evaluation.

The results (Figure 7.17) showed that the feedback was mostly positive. The risk-reduction plot was preferred because this visually provided a sequence of modifications while intuitively presenting the important data of each modification in the same view. When comparing modifications or analyzing patterns, using the risk-reduction plot is more intuitive than investigating data in separate views. The bars for the importance of BEs (on top of the risk reduction plot) also had good reviews because they were easy to understand and dynamically linked to the visualizations of design modifications.

Opinion was somewhat divided on the risk-state node visualizing the modification data. Most complaints were concentrated on the small size of the node. The graphical properties attached to the node are too small. A suggestion was to apply an interactive fish-eye zoom to the node of interest. A participant commented that a risk-state node looked crowded because there are many additional visual items attached to the node. For example, although some attributes of modifications (i.e., the modification cost, modification type, and modification value) provided significant in-

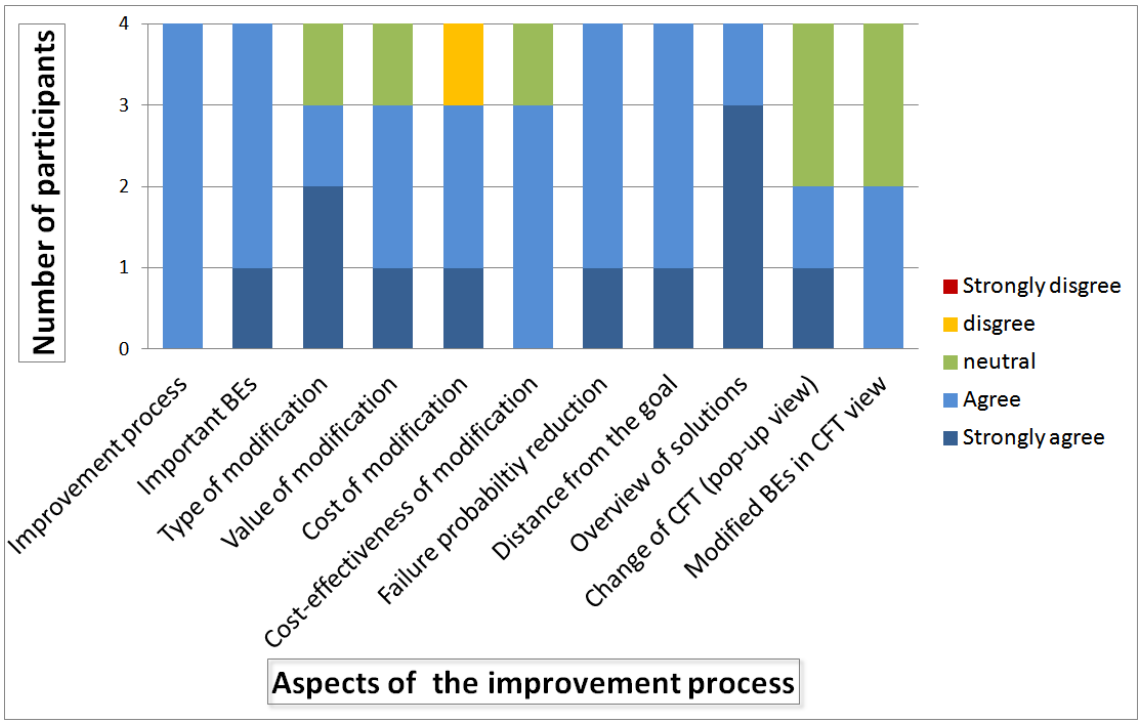


Figure 7.17: Qualitative evaluation.

formation for analysis of the existing modifications, the graphical representations of the data did not play an important role in the construction of a modification. He suggested to dynamically represent the data: show specific graphic properties only when requested.

The representations for the effects of modifications received good comments. The participants could clearly understand how much the risk was reduced by applying a modification and how much the actual risk still needed to be reduced. The participants also positively commented on the overview plots of the solutions. Regarding the adaptation of the CFT structure, the participants thought that the views for showing the CFT structure small, for both the pop-up view and the main CFT view.

In short, the invited domain experts preferred our approach because they believed that the proposed visualizations and interactions may effectively facilitate the identification and analysis of the improvement solutions.

Chapter 8

Framework

The visualization concepts proposed in this dissertation were implemented by developing a visualization system ViSSaAn (Visual Support for Safety Analysis) [209] using Java and the Prefuse library [77]. ViSSaAn accepts data generated by ESSaRel [193] that contains CFT models and textual results of the MCS analysis. Java Swing is used for most of the views and the node-link layouts are implemented with the help of the Prefuse library. ViSSaAn consists of three main parts: the **MCS Matrix**, the **VisQSA system**, and the **General configuration view**.

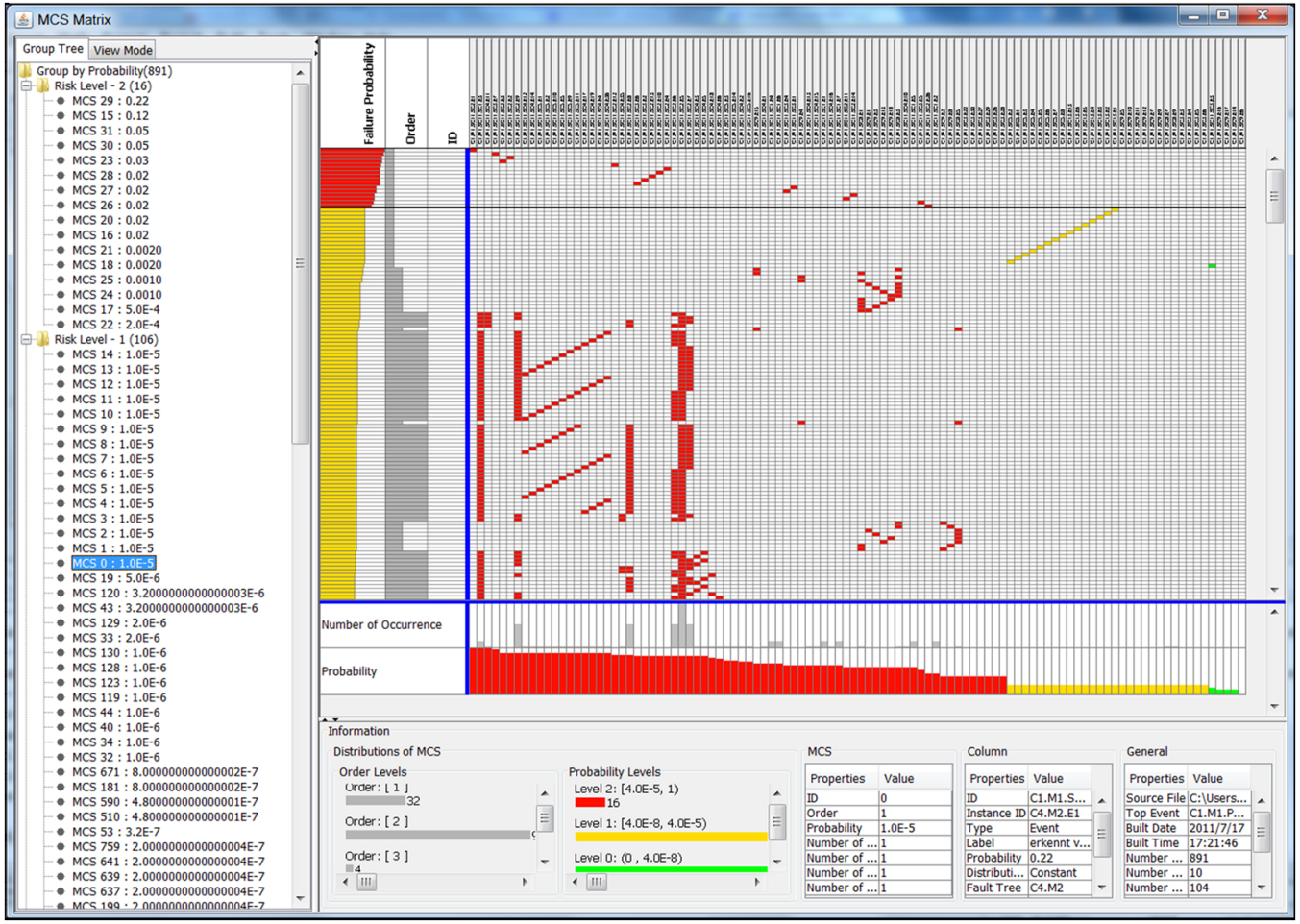
8.1 MCS Matrix System

MCS Matrix introduced in section 5 contains three parts (Figure 8.1 (a)). The central part is the matrix view. The left part is an interaction panel consisting of two tabs: a tree structure for navigating the grouped MCSs and a view where engineers may control most interactions (this view is shown in Figure 8.1 (b)). The information panel (the bottom part of Figure 8.1) displays statistical information, instant data, and the general information of the applied dataset.

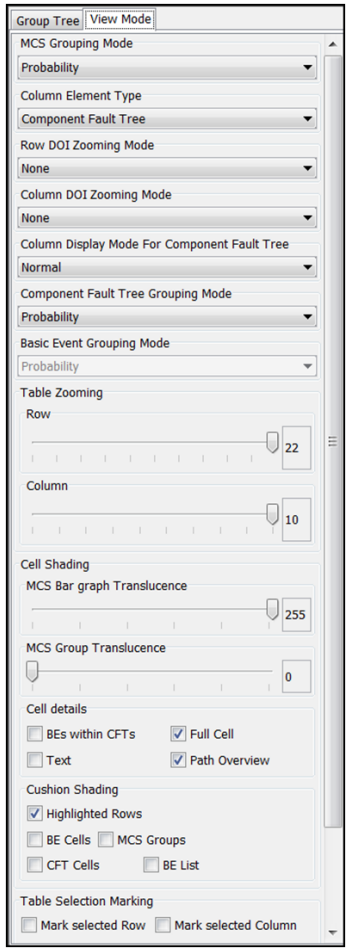
8.2 VisQSA System

The VisQSA system provides the visual quantitative analysis based on the CFT. This system supports the visualization of the importance analysis introduced in Chapter 6, and the visualization of the safety improvement process described in Chapter 7.

Figure 8.2 depicts the main view of VisQSA that contains the CFT view (section 6), the importance plot (section 6.1.5), the risk reduction plot (section 7.1), and the auxiliary overview plots (section 7.1.3). Figure 8.3 shows the overview of the VisQSA system including the main view and additional views. A data table presents the quantitative data of the importance of basic events with filtering and sorting functions (Figure 8.3 (b)). An additional plot where the x-axis represents the failure probabilities of basic events and the y-axis represents the importance of basic events (Figure 8.3 (c)). This plot supports to explore the relations between the failure probability and the importance.



(a) MCS Matrix. The central part is the matrix view; the left part is an interaction panel that consists of a tab of the navigation tree and a tab of the view for controlling the visualizations (shown in (b)); the bottom part is an information panel.



(b) Tab of the control view

Figure 8.1: MCS Matrix system.

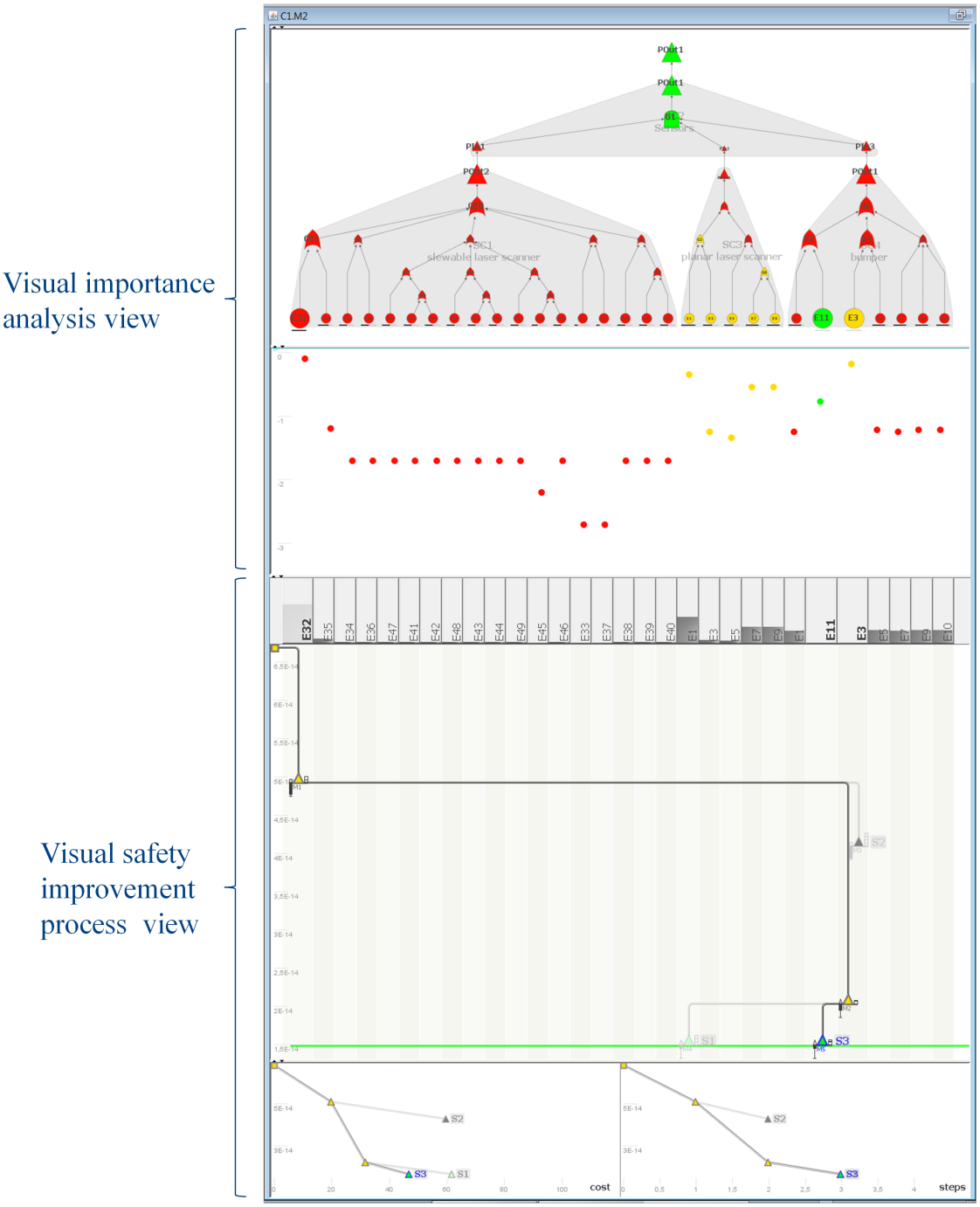


Figure 8.2: Main analysis view of VisQSA. The upper part is the visualization of the importance analysis. The lower part implements the visual safety improvement process.

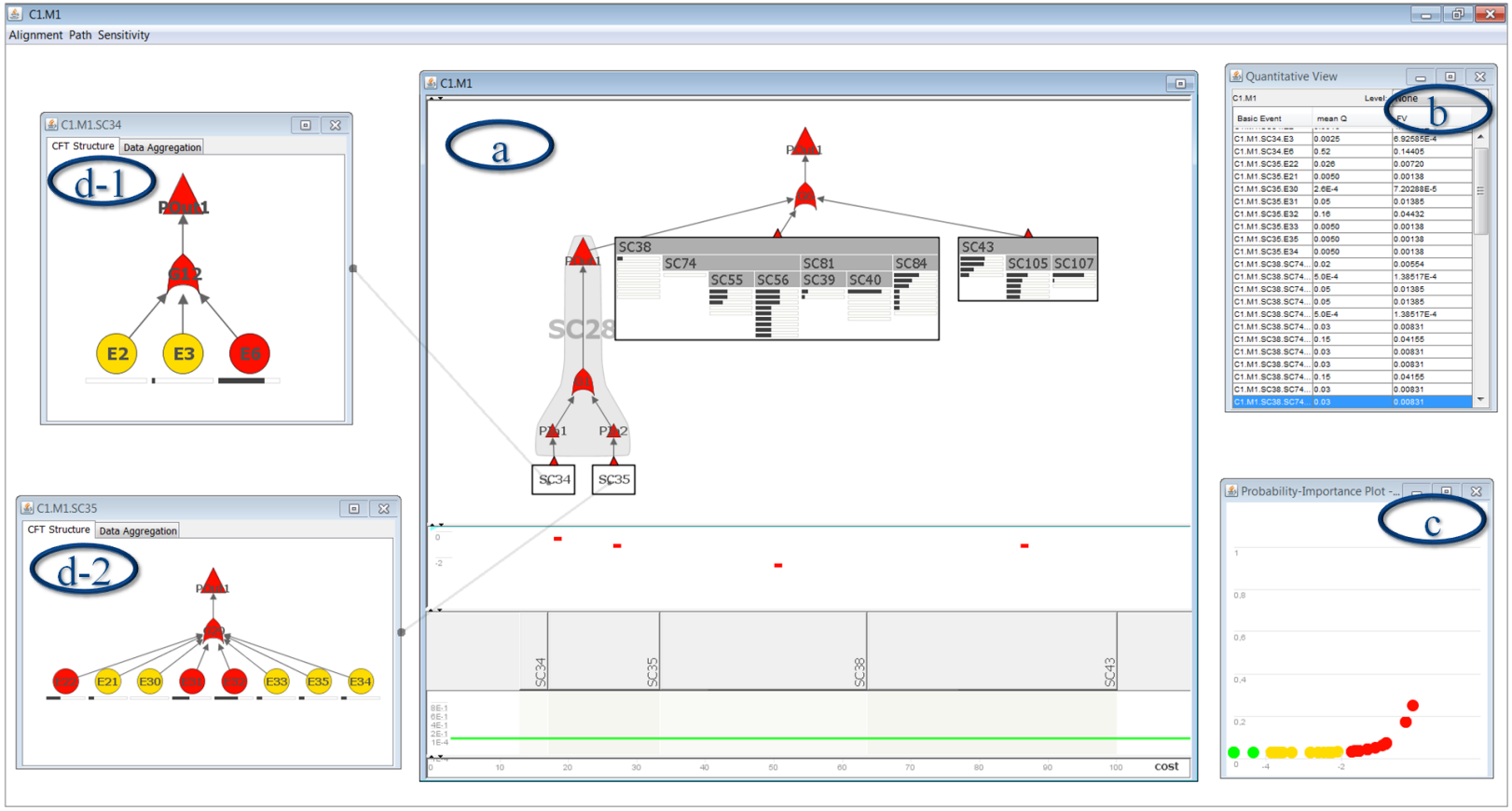


Figure 8.3: Overview of the VisQSA system. (a) main analysis view. (b) data table for the importance of basic events. (c) plot of the failure probability of basic events (x-axis) and the importance (y-axis). (d) separate views representing the logical structures of the desired CFT components.

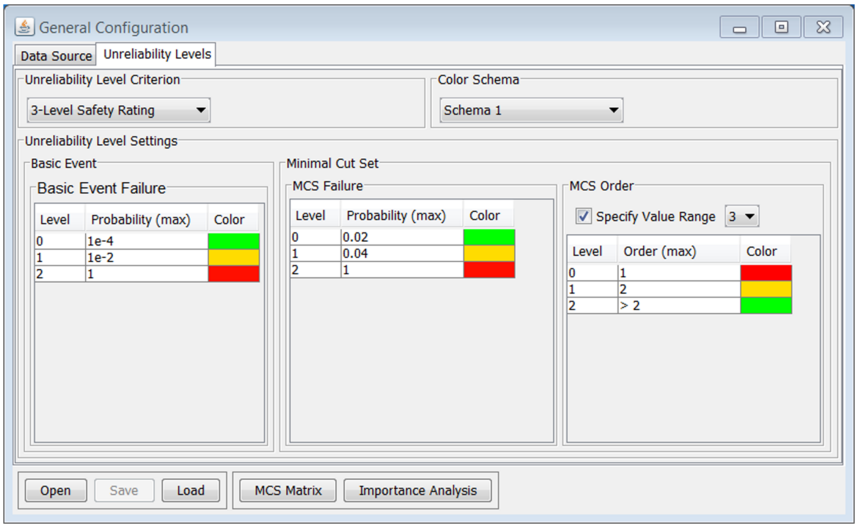


Figure 8.4: General Configuration View. Unreliability levels and their colors may be configured in the view.

When showing logical structures of multiple CFT components, the in-place expansion may make the view look too complicated. The conciseness of the original CFT structure is missing. In this case, engineers look to view CFTs by mixing the in-place expansion concept and the ordinary separate view concept. The traditional concept using separate views is retained in VisQSA (Figure 8.3 (c)). Gray lines link the separate external views to the rectangular symbols that represent the CFT components in the main CFT view. Engineers may analyze the integrated and continuous structure of the most interesting CFT components using the in-place expansion while maintaining the logical structures of other CFT components in external views.

8.3 Configuration View

The starting point of ViSSaAn is a general configuration view (Figure 8.4). The CFTs and MCSs generated by ESSaRel are loaded to ViSSaAn system. Engineers are allowed to set the unreliability levels, and assign colors to the levels. After the configuration of ViSSaAn, the sub-visualization systems, i.e., MCS Matrix and VisQSA, can be undertaken.

Chapter 9

Conclusion

The FTA is widely used to analyze the safety of embedded systems. The CFT is an advanced modeling concept of the FTA. There are qualitative and quantitative analysis concepts based on the FTA. The previous research work focused on the improvements with respect to algorithms and processes. However, the representation concepts are not efficient and effective enough for supporting the complex situations. Our research work focuses on facilitating the analyses based on the CFT by enhancing the representations using suitable visualization methods and flexible interaction techniques. We propose a visualization system called ViSSaAn to implement our approaches that consists of the sub-system MCS Matrix facilitating the MCS analysis and the sub-system VisQSA facilitating the importance analysis and the safety improvement process in visual ways.

9.1 Visualization of MCS Analysis

9.1.1 Summary of Contributions

MCS Matrix is aimed at facilitating the representation of the MCS analysis. It provides a matrix view to represent the relations between MCSs and basic events. Colors are used to encode the criticality of the failure probability of elements of the FT. In order to facilitate the investigation of the significant MCSs, MCs Matrix allows to respectively aggregate MCSs and basic events using flexible grouping concepts. MCS Matrix provides scaling concepts for handling large MCS data. Uniform scaling and the individual scaling may cooperate to represent the detail information while maintaining a satisfactory overview of the large MCS data. The scaling by groups displays different levels of details of the information associated with MCSs (or basic events) according to the unreliability levels of MCSs (basic events). In this way, the limited screen space can be effectively used for primarily representing the significant information. In order to effectively represent the failure propagation of MCSs, MCS Matrix provides a concept that integrates the failure flow of MCSs with the matrix view.

In general, MCS Matrix effectively represents the relations between MCSs and basic events. An overview of a large set of MCSs may be maintained when investigating detailed information associated with MCSs. Engineers can easily identify

the significant information with the help of information highlighting methods. The hidden information, e.g., patterns of MCSs and relation between basic events, may be discovered by using our visualization. Engineers may investigate the influence of MCSs along the CFT structure directly in the matrix view. This greatly improve efficiency of the MCS analysis in visual ways.

9.1.2 Future Work

For the current MCS matrix, the scaling by groups (Figure 5.7) may not provide sufficient benefit because plenty of critical MCSs may take up a great deal of screen space. In the future, we need a way to reduce the space of the critical MCSs and show as much important information of these MCSs as possible. A possible solution is the fish-eye technique. Another possible extension is the logical relations between CFT components. Currently, we individually show the structure of the specific CFT component in the embedded view of the matrix layout. In the future, we may focus on simultaneously representing structures of multiple CFT components in different embedded views without great increase of space requirement and show the logical data flow between these components.

9.2 Visualization of Importance Analysis

9.2.1 Summary of Contributions

In the dissertation we propose a visualization method that represents insights for the importance analysis of CFTs by composing the iceray tree and the node-link graph. The iceray tree view is responsible for representing the overview of the importance of basic events by taking the hierarchical architecture in system design into account while the node-link diagram focuses on representing the CFT structure. In order to effectively use the screen space, the structure of CFT components are visible only when requesting by either referencing the deeply nested CFT components or regrading the continuous critical paths. In order to indicate the failure propagation of the important basic events, the critical paths are highlighted using border colors. The translucent gray blobs are used to enclose the nodes of each expanded CFT components in order to identify the CFT components. To facilitate the comparison of basic events, we provide the aggregate method and alignment layout concepts to visually enhance the CFT structure. We also provide an additional importance plot to quickly compare a large number of basic events. In short, using our visualization method, engineers may estimate and compare the importance of basic events while analyzing the logical structure of the CFT. Additionally, engineers may quickly investigate the basic events in any CFT component by using the architectural view of our method without being disturbed by complex nesting relations between components.

9.2.2 Future Work

There are still some meaningful work to improve the visualization of the importance analysis in the future. A significant issue of our approach is that the CFT structure takes up a lot of space (Figure 6.13) even though it provides an intuitive structure for the failure flow and logical relations between the important basic events. The critical path of the failure flow may only be a small part of the whole logical structure of a CFT component. To address this issue, we need a mechanism that represents the failure flow directly in the architectural view rather than along the complete logical structure. We discussed this possibility in section 4.2.3.2 when designing our visualization. Considering the intuition issue and the visual clutter problems, we rejected this idea. However, as a complement, the embedded failure flow still makes sense. We may show the simple failure flow of only a single specific basic event in the architectural view in order to reduce the influence on the architectural view.

Our work currently focuses on only one importance analysis method, i.e., the Fussell-Vesely measure. In the future, more measures may be integrated into our visualization. In this way, basic events may be estimated and ranked according to different metrics that are respectively suitable for various analysis demands. To refine our visualization, a better algorithm for the blobs is desired to address the overlapping issue and a more effective algorithm is needed for reducing the line-crossing of the hierarchical logical structure of the CFT. In addition, the interactive association between the MCS Matrix and the VisQSA will be a meaningful work. This way, it may be possible to dynamically and smoothly switch views between qualitative measure regarding the MCSs analysis and the quantitative measure regarding the importance analysis in our visualization system.

9.3 Visualization of the Safety Improvement Process

9.3.1 Summary of Contributions

To facilitate the iterative safety improvement process, we propose a visualization method that provides a risk-reduction plot to integrate the significant data that is distributed in separate views which hampers the efficient identification of solutions. The risk-reduction plot is a combination between a decision tree representing the identifying process of modifications and a scatter plot that represents the cause-effect relation between the safety vulnerability related to modifications and the corresponding improvement results. The decision tree links the possible modifications to form improvement solutions. The risk-reduction plot allows engineers to analyze the essential data of the modifications of interest while maintaining the overall sequence of modifications for representing the identifying process of modifications. This is helpful for analyzing and optimizing existing solutions and also has meaning for new design modifications. Our visualization allows engineers to dynamically investigate the change of the structure of CFT components with a pop-up view. The visual safety improvement process also provides the overviews of the solutions with respect to different aspects in auxiliary plots. In this case, engineers are allowed to

identify optimal solution(s) from a large set of solutions with respect to the cost or the modification steps. In general, the proposed visual safety improvement process allows engineers to visually and interactively identify a series of design modifications, and facilitates the effective determination of the optimal modifications and solutions with the help of appropriate graphical properties.

9.3.2 Future Work

An unresolved issue of our visualization is the alignment between the CFT structure view and the risk-reduction plot. When all basic events of a CFT are shown in the CFT view, the basic event node can be perfectly aligned with the x-axis of the risk-reduction plot (Figure 7.10). However, if the structure of a CFT component is not shown, the alignment will not work very well (Figure 7.14). We need to improve the algorithm of the alignment in future. Additionally, dynamically showing the graphical properties of the risk-state node is also a meaningful improvement to the visualization. This method may help to clearly present a large number of risk-state nodes on the screen.

Bibliography

[1] J. Abello and F. van Ham. Matrix zoom: A visual interface to semi-external graphs. In *INFOVIS*, pages 183–190. IEEE Computer Society, 2004.

[2] Food Standards Agency. Traffic light labelling, 2009.

[3] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, pages 619–626, 1992.

[4] Y. I. Al-Zokari, T. Khan, D. Schneider, D. Zeckzer, and H. Hagen. CakES: Cake Metaphor for Analyzing Safety Issues of Embedded Systems. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 1–16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

[5] Y. I. Al-Zokari, D. Schneider, D. Zeckzer, L. Guzman, Y. Livnat, and H. Hagen. Enhanced CakES representing Safety Analysis results of Embedded Systems. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *FedCSIS*, pages 783–790, 2011.

[6] ALD. RAM Commander. `http://www.aldservice.com`, Online; accessed 15-Jun-2012.

[7] K. Andrews and H. Heidegger. Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs. In *Proc. of Infovis '98*, 1998.

[8] M. Ankerst, C. Elsen, M. Ester, and H-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining (KDD'99)*, pages 392–396, 1999.

[9] M. Ankerst, M. Ester, and H-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 179–188, 2000.

[10] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, 2004.

[11] E. H. Baehrecke, N. Dang, K. Babaria, and B. Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *Bmc Bioinformatics*, 5, 2004.

[12] M. Balzer and O. Deussen. Voronoi Treemaps. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 7–, Washington, DC, USA, 2005.

[13] T. Barlow and P. Neville. A Comparison of 2-D Visualizations of Hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 131–138. IEEE Computer Society, 2001.

[14] T. Barlow and P. Neville. Case Study: Visualization for Decision Tree Analysis in Data Mining. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 149–152. IEEE Computer Society, 2001.

[15] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara. Drawing Directed Acyclic Graphs: An Experimental Study. Technical report, Department of Computer Science, Brown University, Providence, RI, USA, 1996.

[16] P. Baudisch, B. Lee, and L. Hanna. Fishnet, a Fisheye Web Browser with Search Term Popouts: A Comparative Evaluation with Overview and Linear View. In *AVI '04: Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 133–140, New York, NY, USA, 2004. ACM.

[17] F. Beck and S. Diehl. Visual comparison of software architectures. In *Proceedings of the 5th international symposium on Software visualization*, SOFTVIS '10, pages 183–192, New York, NY, USA, 2010. ACM.

[18] R. A. Becker, S. G. Eick, and A. R. Wilks. Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):16–28, March 1995.

[19] B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. A Zoomable Graphical Interface for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 1:3–31, 7 1996.

[20] B. B. Bederson, J. Meyer, and L. Good. Jazz: an extensible zoomable user interface graphics toolkit in Java. In *Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST'2000)*, pages 171–180, San Diego, USA, 2000.

[21] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, October 2002.

[22] J. Bertin. *Graphics and Graphic Information Processing*. Walter de Gruyter & Co, Berlin, 1967. Berg, William J. and Scott, Paul, editors. Translated into English 1981 from "La graphique et le traitemente graphique de l'information".

[23] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.

[24] J. Bertin. Matrix theory of graphics. *Information Design Journal*, pages 5–19, 2001.

[25] Z. W. Birnbaum. On the importance of different components in a multicomponent system. *Multivariate analysis*, 1969.

[26] S. Böttger, H. Barthel, and A. Ebert. Fault forest visualization. In *Proceedings of the 5th international symposium on Software visualization (SOFTVIS '10)*, Salt Lake City, Utah, USA, 2010.

[27] F. Boutin, J. Thievre, and M. Hascoeet. Focus-based filtering + clustering technique for power-law networks with small world phenomenon. In *Proceedings of SPIE*, 6060, pages 236–247, 2006.

[28] F. Boutin, J. Thièvre, and M. Hascoët. Multilevel compound tree: construction visualization and interaction. In *Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*, INTERACT'05, pages 847–860, Berlin, Heidelberg, 2005. Springer-Verlag.

[29] M. Bozzano and A. Villafiorita. The FSAP/NuSMV-SA Safety Analysis Platform. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(1):5–24, 2007.

[30] M. Bruls, K. Huizing, and J. J. van Wijk. Squarified Treemaps. In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42, 1999.

[31] M. Burch, F. Bott, F. Beck, and S. Diehl. Cartesian vs. radial — a comparative evaluation of two visualization tools. In *Proceedings of the 4th International Symposium on Advances in Visual Computing (ISVC '08)*, pages 151–160. Springer-Verlag, 2008.

[32] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17:2440–2448, December 2011.

[33] H. Byelas and A. Telea. Visualizing metrics on areas of interest in software architecture diagrams. In *Proceedings of the 2009 IEEE Pacific Visualization Symposium*, PACIFICVIS '09, pages 33–40, Washington, DC, USA, 2009. IEEE Computer Society.

[34] A. C. Caputo, M. Palumbo, and R. Tartaglia. Fault Tree Analysis for Risk Assessment in the Borexino Experiment. In *Process Safety Progress*, volume 23, pages 121–131. American Institute of Chemical Engineers, June 2004.

[35] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[36] D. B. Carr, R. J. Littlefield, and W. L. Nichloson. Scatterplot matrix techniques for large N. In *Proceedings of the Seventeenth Symposium on the interface of computer sciences and statistics on Computer science and statistics*, pages 297–306, New York, NY, USA, 1986. Elsevier North-Holland, Inc.

[37] CENELEC. EN50126: Railway applications. The specification and demonstration of reliability, availability, maintainability and safety (RAMS), 1999.

[38] M. C. Cheok, G. W. Parry, and R. R. Sherry. Use of importance measures in risk-informed regulatory applications. *Reliability Engineering and System Safety*, 60(3):213 – 226, 1998.

[39] F. Chevalier, D. Auber, and A. Telea. Structural analysis and visualization of C++ code evolution using syntax trees. In *Ninth international workshop on Principles of software evolution: in conjunction with the 6th ESEC/FSE joint meeting*, IWPSE '07, pages 90–97, New York, NY, USA, 2007. ACM.

[40] Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization 2000*, INFOVIS '00, Washington, DC, USA, 2000. IEEE Computer Society.

[41] Ed H. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Information Visualization Symposium (InfoVis '98)*, 1998.

[42] R. D. Cigolini, A.V. Deshmukh, L. Fedele, and S.A. McComb. *Recent Advances in Maintenance and Infrastructure Management.* Springer-Verlag, 2009.

[43] E. M. Clarke and J. M. Wing. Formal methods: state of the art and future directions. *ACM Comput. Surv.*, 28(4):626–643, December 1996.

[44] A. Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *Proceedings of the fifth conference on Australasian user interface*, volume 28 of *AUIC '04*, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[45] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM COMPUT. SURV*, pages 1–31, 2008.

[46] A. Cockburn and B. McKenzie. 3d or not 3d?: evaluating the effect of the third dimension in a document management system. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '01)*, pages 434–441, New York, NY, USA, 2001. ACM.

[47] C. Collins, G. Penn, and S. Carpendale. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1009–1016, oct 2009.

[48] S. Contini, L. Fabbri, and V. Matuzas. Concurrent Importance and Sensitivity Analysis Applied to Multiple Fault Trees. *JRC IPSC report, EUR 23825 EN, Ispra.*, 2009.

[49] S. Contini, L. Fabbri, and V. Matuzas. A novel method to apply Importance and Sensitivity Analysis to multiple Fault-trees. *Journal of Loss Prevention in the Process Industries*, 3, 2010.

[50] S. Contini, S. Scheer, and M. Wilikens. Sensitivity Analysis for System Design Improvement. In *Proceedings of the 2000 International Conference on Dependable Systems and Networks*, DSN '00, pages 243–248. IEEE Computer Society, 2000.

[51] R. B. Cross and J. E. Ballesio. An Integrated Quantitative Risk Assessment of an Oil Carrier. In *Safety and reliability: proceedings of ESREL 2003, European Safety and Reliability Conference 2003, Maastricht, The Netherlands*, 2003.

[52] W. De Pauw, D. Kimelman, and J. Vlissides. Modeling Object-Oriented Program Execution. *ECOOP94 Conference Proceedings*, pages 163–182, 1994.

[53] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4(5):235–282, October 1994.

[54] S. Diehl, F. Beck, and M. Burch. Uncovering strengths and weaknesses of radial visualizationsan empirical approach. *IEEE Transactions on Visualization and Computer Graphics*, 16, 2010.

[55] DIN. Fehlerbaumanalyse (Fault Tree Analysis). *German Industry Standard (Part 1 & 2)*, 1981//1990. Beuth Verlag, Berlin.

[56] DIN. DIN EN ISO 9126-1. *Software engineering Product quality*, 2001.

[57] U. Dogrusöz, B. Madden, and P. Madden. Circular Layout in the Graph Layout Toolkit. In *Proceedings of the Symposium on Graph Drawing*, GD '96, pages 92–100, London, UK, 1997. Springer-Verlag.

[58] G. M. Draper, Y. Livnat, and R. F. Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15:759–776, September 2009.

[59] P. Eades. *Drawing free trees*, volume 5. Bulletin of the Institute of Combinatorics and its Applications, 1992.

[60] P. A. Eades. A heuristic for graph drawing. In *Congressus Numerantium*, volume 42, pages 149–160, 1984.

[61] N. Elmqvist, T-N. Do, H. Goodell, N. Henry, and J-D. Fekete. ZAME: Interactive Large-Scale Graph Visualization. In *PacificVis*, pages 215–222. IEEE, 2008.

[62] N. Elmqvist and J-D. Fekete. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, May 2010.

[63] C. Ericson. Fault Tree Analysis - A History. In *Proceedings of the 17th International Systems Safety Conference*, 1999.

[64] N. S. Fard. Determination of minimal cut sets of a complex fault tree. *Computers & Industrial Engineering*, 33(1-2):59–62, 1997. Proceedings of the 21st International Conference on Computers and Industrial Engineering.

[65] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Softw. Pract. Exper.*, 21(11):1129–1164, November 1991.

[66] G. W. Furnas. Generalized fisheye views. In *Proceedings of the ACMSIGCHI Conference on Human Factors in Computing Systems*, pages 16–23. ACM, 1986.

[67] J. B. Fussel and W. E. Vesely. A New Methodology For Obtaining Cut Sets From Fault Trees. *ANS Trans*, 15, 1972.

[68] J. B. Fussell. How to hand calculate system reliability characteristics. *IEEE Transactions on Reliability*, R-24:169–174, 1975.

[69] M. Ghoniem, J-D. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 17–24, Washington, DC, USA, 2004. IEEE Computer Society.

[70] M. Ghoniem, J-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4:114–135, July 2005.

[71] M. Graham and J. Kennedy. Exploring and examining assessment data via a matrix visualization. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 158–162, New York, NY, USA, 2004. ACM.

[72] M. C. Hao, U. Dayal, D. Keim, and T. Schreck. A visual analysis of multi-attribute data using. *Pixel Matrix Displays, Proc. IS& T/SPIE Conference on Visualization and Data Analysis*, 2007.

[73] M. C. Hao, U. Dayal, D. A. Keim, and T. Schreck. Importance-driven visualization layouts for large time series data. *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 203–210, 2005.

[74] M. C. Hao, U. Dayal, D. A. Keim, and T. Schreck. Multi-resolution techniques for visual exploration of large time-series data. In K. Museth, T. Möller, and A. Ynnerman, editors, *EuroVis*, pages 27–34. Eurographics Association, 2007.

[75] M. C. Hao, M. Hsu, U. Dayal, and A. Krug. Web-Based Visualization of Large Hierarchical Graphs Using Invisible Links in a Hyperbolic Space. In *Proceedings of the Fifth Working Conference on Visual Database Systems: Advances in Visual Information Management*, VDB 5, pages 83–94, Deventer, The Netherlands., 2000. Kluwer, B.V.

[76] D. Harel. Statecharts: A Visual Formulation for Complex Systems. *Science of Computer Programming*, 8:231–274, 1987.

[77] J. Heer. Prefuse. `http://prefuse.org/`, Online; accessed 15-Jun-2012.

[78] J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 23–25, Washington, DC, USA, 2005. IEEE Computer Society.

[79] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective Visualization of File System Access-Control. In *Proceedings of the 5th international workshop on Visualization for Computer Security*, VizSec '08, Berlin, Heidelberg, 2008. Springer-Verlag.

[80] N. Henry and J-D. Fekete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 12:677–684, 2006.

[81] N. Henry and J-D. Fekete. MatLink: enhanced matrix visualization for analyzing social networks. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, INTERACT'07, pages 288–302, Berlin, Heidelberg, 2007. Springer-Verlag.

[82] N. Henry, J-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, November 2007.

[83] R. D. Herbert, R. Webber, and W. Jiang. Space-filling Techniques in Visualizing Output from Computer Based Economic Models. *12th International Conference on. Computing in Economics and Finance*, 2006.

[84] I. Herman, G. Melancon, M. M. Ruiter, and M. Delest. *Latour - a tree visualisation system.* CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, 1999.

[85] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *Proceedings of the 8th conference on Visualization 1997 (VIS '97)*, pages 437–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[86] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12:2006, 2006.

[87] D. Holten, R. Vliegen, and J. J. van Wijk. Visual realism for the visualization of software metrics. In *VISSOFT05: Proceedings of 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis (2005), IEEE CS*, pages 27–32. IEEE Computer Society, 2005.

[88] HSE. Health and Safety Executive. `http://www.hse.gov.uk/`, Online; accessed 23-Jun-2012.

[89] IAEA. Case study on the use of PSA methods: determining safety importance of systems and components at nuclear power plants. *IAEA TECDOC 590*, 1991.

[90] IEC. International Electrotechnical Vocabulary.

[91] IEC. Fault Tree Analysis. *International Standard IEC 61025*, 1990. Geneva.

[92] IEC. Functional safety of electrical/electronic/programmable electronic safety-related systems. *International Standard IEC 61508*, 2000.

[93] IEC. Risk management-Risk assessment techniques. *International Standard IEC 31010*, 2009.

[94] Clifton A. E. II and The Boeing Company. Fault Tree Analysis By Design. *Proc. 16th International System Safety Conference*, 1998.

[95] R. L. Iman. A matrix-based approach to uncertainty and sensitivity analysis for fault trees. *Risk Analysis*, 7(1), 1987.

[96] ISO. Quality management and quality assurance - Vocabulary. *DIN EN ISO 8402*, 1994.

[97] ISO. ISO 17666. *Space Systems Risk Management*, 2001.

[98] ISO. ISO 9000. *Quality management systems - Fundamentals and Vocabulary*, 2005.

[99] ISO. ISO 31000. *Risk management – Principles and guidelines*, 2009.

[100] ISOGRAPH. FaultTree+. `http://www.isograph-software.com`, Online; accessed 15-Mar-2012.

[101] ITEM-Software. ITEM ToolKit. `http://www.itemtoolkit.com/`, Online; accessed 15-Jun-2012.

[102] T. J. Jankun-Kelly and K-L. Ma. MoireGraphs: radial focus+context visualization and interaction for graphs with visual nodes. In *Proceedings of the Ninth annual IEEE conference on Information visualization*, INFOVIS'03, pages 59–66, Washington, DC, USA, 2003. IEEE Computer Society.

[103] L. Jin and D. C. Banks. Tennisviewer: A browser for competition trees. *IEEE Comput. Graph. Appl.*, 17(4):63–65, July 1997.

[104] M. John, C. Tominski, and H. Schumann. Visual and analytical extensions for the table lens. In *Visualization and Data Analysis 2008*, volume 6809 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, 2008.

[105] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. *Visualization '91, Proceedings., IEEE Conference on*, pages 284–291, 1991.

[106] W-A. Jungmeister and D. Turo. Adapting Treemaps To Stock Portfolio Visualization. *Center for Automation Research Technical Report, University of Maryland*, 1992.

[107] B. Kaiser. A fault-tree semantics to model software-controlled systems. *Softwaretechnik-Trends 23(3) Gesellschaft fuer Informatik (Hg.)*, 2003.

[108] B. Kaiser, P. Liggesmeyer, and O. Maekel. A New Component Concept for Fault Trees. In *Proceedings of the 8th Australian workshop on safety critical systems and software (SCS' 03)*, 2003.

[109] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, April 1989.

[110] C. Kara-Zaitri. An improved minimal cut set algorithm. *International Journal of Quality & Reliability Management*, 13, 1996.

[111] D. Kececioglu. *Reliability Engineering Handbook*, volume 2. DEStech Publications, Inc, 1991.

[112] D. A. Keim, F. Mansmann, J. Schneidewind, and T. Schreck. Monitoring network traffic with radial traffic analyzer. In *In 2006 IEEE Symposium On Visual Analytics Science And Technology (VAST)*, pages 123–128, 2006.

[113] R. Keller, C. M. Eckert, and P. J. Clarkson. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization*, 5:62–76, March 2006.

[114] R. Kincaid. VistaClara: an interactive visualization for exploratory analysis of DNA microarrays. In *Proceedings of the 2004 ACM symposium on Applied computing*, SAC '04, pages 167–174, New York, NY, USA, 2004. ACM.

[115] R. Kincaid. Line graph explorer: scalable display of line graphs using focus+context. In *In Working Conference on Advanced Visual interfaces*, pages 404–411. ACM Press, 2006.

[116] A. Kjellin, L. W. Pettersson, S. Seipel, and M. Lind. Different levels of 3d: an evaluation of visualized discrete spatiotemporal data in space-time cubes. *Information Visualization*, 9:152–164, June 2010.

[117] B. Kleiner and J. A. Hartigan. Representing points in many dimensions by trees and castles. *Journal of the American Statistical Association*, 76(374):260–269, 1981.

[118] J. Knight. Safety critical systems: challenges and directions. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 547–550, New York, NY, USA, 2002. ACM.

[119] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1st edition, 1997.

[120] R. Kosara. Visualization criticism - the missing link between information visualization and art. In *Proceedings of the 11th International Conference Information Visualization*, pages 631–636, Washington, DC, USA, 2007. IEEE Computer Society.

[121] J. B. Kruskal and J. M. Landwehr. Icicle Plots: Better Displays for Hierarchical Clustering. *The American Statistician*, 37(2):162–168, 1983.

[122] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[123] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veksler, W. D. Gray, and C. Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, November 2006.

[124] J. C. Lee and N. J. McCormick. *Risk and Safety Analysis of Nuclear Systems*. Wiley, 2011.

[125] E. Levy, J. Zacks, B. Tversky, and D. Schiano. Gratuitous graphics? putting preferences in perspective. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, CHI '96, pages 42–49, New York, NY, USA, 1996. ACM.

[126] Liggesmeyer, P. Lecture: Safety and Reliability of Embedded Systems, University of Kaiserslautern, 2011. `http://agde.informatik.uni-kl.de/teaching/suze/ws2011/material/folien/`, Online; accessed 15-Jun-2012.

[127] C-C. Lin and H-C. Yen. On balloon drawings of rooted trees. In *Proceedings of the 13th international conference on Graph Drawing*, GD'05, pages 285–296, Berlin, Heidelberg, 2006. Springer-Verlag.

[128] Y. Liu and G. Salvendy. Interactive visual decision tree classification. In *Proceedings of the 12th international conference on Human-computer interaction: interaction platforms and techniques (HCI'07)*, pages 92–105. Springer-Verlag, 2007.

[129] R. G. Lomax. *Statistical Concepts: A Second Course*. Routledge Academic, 2007.

[130] G. Lommerse, F. Nossin, L. Voinea, and A. Telea. The Visual Code Navigator: An Interactive Toolset for Source Code Investigation. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 4–, Washington, DC, USA, 2005. IEEE Computer Society.

[131] H. Lü and J. Fogarty. Cascaded treemaps: examining the visibility and stability of structure in treemaps. In *Proceedings of graphics interface 2008*, GI '08, pages 259–266, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[132] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5:110–141, 1986.

[133] M. J. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 16–23, oct. 2005.

[134] M. J. McGuffin and J-M. Robert. Quantifying the space-efficiency of 2D graphical representations of trees. *Information Visualization*, 9(2):115–140, June 2010.

[135] G. Melancon and I. Herman. *Circular drawings of rooted trees*. CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, 1998.

[136] K. B. Misra. *Handbook of Performability Engineering*. Springer-Verlag, 2008.

[137] M. Modarres. *Risk Analysis in Engineering: Techniques, Tools, and Trends*. CRC, 1 edition, 2006.

[138] J. Murtha. Evindence theory and fault tree analysis to cost-effectively improve reliability in small UAV design. *Virginia Space Grant Consortium Student Research Conference*, 2009.

[139] K. Nakashima and Y. Hattori. An efficient bottom-up algorithm for enumerating minimal cut sets of fault trees. IEEE Trans. Reliab., R-28, (5) 353 (December 1979). *Microelectronics and Reliability*, 20(4):543–543, 1980.

[140] NASA. Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners. *NASA*, 2002.

[141] NASA. Risk Management Reporting. *GSFC-STD-0002*, 2009.

[142] NASA, W. E. Vesely, J. Dugan, J. Fragola, J. MinarickIII, J. Railsback, and M. Stamatelatos. Fault Tree Handbook with Aerospace Applications. *NASA*, 2002.

[143] NASA, W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. Fault Tree Handbook. *U.S. Nuclear Regulatory Commission*, 1981.

[144] P. Neumann, S. Schlechtweg, and S. Carpendale. ArcTrees: Visualizing Relations in Hierarchical Data. In Ken W. Brodlie, David J. Duke, and Ken I. Joy, editors, *Data Visualization 2005, Eurographics/IEEE VGTC Symposium on Visualization Symposium Proceedings*, pages 53–60, Aire-la-Ville, Switzerland, 2005. The Eurographics Association.

[145] T. D. Nguyen, T. B. Ho, and H. Shimodaira. A visualization tool for interactive learning of large decision trees. In *Tools with Artificial Intelligence, 2000. ICTAI 2000. Proceedings. 12th IEEE International Conference on.*, pages 28–35, 2000.

[146] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim. Visual readability analysis: How to make your writings easier to read. *IEEE Transactions on Visualization and Computer Graphics*, 99, 2011.

[147] United States Department of Defense. *Risk Management Guide for DoD Acquisition*. General Books LLC, 2011.

[148] G. M. Olson and J. S. Olson. Groupware and computer-supported cooperative work. In J. A. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 583–595. Lawrence Erlbaum Associates, Inc., 2003.

[149] Y. Ou and J. B. Dugan. Sensitivity Analysis of Modular Dynamic Fault Trees. In *Proceedings of the 4th International Computer Performance and Dependability Symposium*, page 35. IEEE Computer Society, 2000.

[150] D. Pantförder and B. Vogel-Heuser. Benefit and evaluation of interactive 3d process data visualization in operator training of plant manufacturing industry. In *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, SMC'09, pages 824–829, Piscataway, NJ, USA, 2009. IEEE Press.

[151] R. M. Patton, J. M. Beaver, C. A. Steed, T. E. Potok, and J. N. Treadwell. Hierarchical clustering and visualization of aggregate cyber data. In *IWCMC*, pages 1287–1291. IEEE, 2011.

[152] N-K. Pham, T-N. Do, F. Poulet, and A. Morin. Interactive Exploration of Decision Tree Results. *Applied Stochastic Model and Data Analysis International Conference, ASMDA'2007*, pages 152–160, 2007.

[153] P. Pirolli and R. Rao. Table lens as a tool for making sense of data. *Proceedings of the workshop on Advanced visual interfaces*, pages 67–80, 1996.

[154] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 57–64, 2002.

[155] Protovis. Protovis. http://mbostock.github.com/protovis//, Online; accessed 15-Jun-2012.

[156] H. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward. Theoretical foundations of information visualization. In A. Kerren, J. Stasko, J-D. Fekete, and C. North, editors, *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, chapter 3, pages 46–64. Springer Berlin / Heidelberg, Berlin, H., 2008.

[157] Randelshofer, W. randelshofer. http://www.randelshofer.ch/treeviz/, Online; accessed 15-Jun-2012.

[158] R. Rao and S. K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '94)*, 1994.

[159] M. Rausand and A. Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications, Second Edition*. Wiley-Interscience, 2003.

[160] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE Trans. Softw. Eng.*, 7(2):223–228, 3 1981.

[161] RELEXSOFTWARE. Relex Architect. http://www.relexsoftware.co.uk, Online; accessed 15-Jun-2012.

[162] RELIASOFT. BlockSim. http://www.reliasoft.com/BlockSim, Online; accessed 15-Mar-2012.

[163] P. Robertson and L. De Ferrari. Systematic approaches to visualization: Is a reference model needed? *Scientific Visualization: Advances and Challenges*, 1994.

[164] Robotics Research Lab. The Robotics Research Lab at the University of Kaiserslautern. http://agrosy.informatik.uni-kl.de, Online; accessed 15-Jun-2012.

[165] S. Rufiange, M. J. McGuffin, and C. P. Fuhrman. TreeMatrix: a hybrid visualization of compound graphs. *Computer Graphics Forum*, 31:89–101, 2012.

[166] M. M. Sebrechts, J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller. Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 3–10, New York, NY, USA, 1999. ACM.

[167] Z. Shen and K-L. Ma. Path Visualization for Adjacency Matrices. In *Proceedings of Eurographics/IEEE VGTC Syposium on Visualization*, pages 83–90, May 2007.

[168] B. Shneiderman. Direct manipulation: A step beyond programming languages. In R. M. Baecker, editor, *Human-computer interaction*, pages 461–467. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[169] B. Shneiderman. Tree visualization with treemaps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, January 1992.

[170] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Visual Languages, IEEE Symposium on*, 0:336–343, 1996.

[171] B. Shneiderman and A. Aris. Network Visualization by Semantic Substrates. *IEEE transaction on visualization and computer graphics*, 12(5), 2006.

[172] B. Shneiderman and M. Wattenberg. Ordered Treemap Layouts. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS '01)*, INFOVIS '01, pages 73–, Washington, DC, USA, 2001. IEEE Computer Society.

[173] M. Sifer. A Filter Co-ordination for Exploring Multi-Classification Sitemaps. In *Proceedings of the conference on Coordinated and Multiple Views In Exploratory Visualization*, pages 112–, Washington, DC, USA, 2003. IEEE Computer Society.

[174] H. Siirtola. Interaction with the Reorderable Matrix. *Information Visualisation, International Conference on*, 0:272, 1999.

[175] H. Siirtola and E. Mäkinen. Constructing and reconstructing the reorderable matrix. *Information Visualization*, 4:32–48, March 2005.

[176] P. Simonetto, D. Auber, and D. Archambault. Fully Automatic Visualisation of Overlapping Sets. *Computer Graphics Forum*, 28(3):967–974, 2009.

[177] H. S. Smallman, M. St. John, H. M. Oonk, and M. B. Cowen. Information availability in 2d and 3d displays. *IEEE Comput. Graph. Appl.*, 21:51–57, September 2001.

[178] A. M. Smith, W. Xu, Y. Sun, J. R. Faeder, and G. E. Marai. RuleBender: Integrated visualization for biochemical rule-based modeling. In *Biological Data Visualization (BioVis), 2011 IEEE Symposium on*, pages 103 –110, oct. 2011.

[179] M. Spenke, C. Beilken, and T. Berlage. FOCUS: the interactive table for product comparison and selection. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 41–50, New York, NY, USA, 1996. ACM Press.

[180] J. Stasko and E. Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 57–65, Washington, DC, USA, 2000. IEEE Computer Society.

[181] S. Subramanian, D. Aliakseyeu, and J-B. Martens. Empirical evaluation of performance in hybrid 3d and 2d interfaces. In *Proceedings of Human-Computer Interaction - INTERACT 03*, pages 916–919. IOS Press, 2003.

[182] K. Sugiyama. *Graph drawing and applications for software and knowledge engineers*. Series on software engineering and knowledge engineering ;11. World Scientific, Singapore, 2002.

[183] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

[184] K. L. Summers. *Visualization of programs using proximity to trigger continuous semantic zooming: an experimental study*. PhD thesis, Department of Computer Science, 2002.

[185] K. L. Summers, T. E. Goldsmith, S. Kubica, and T. P. Caudell. An experimental evaluation of continuous semantic zooming in program visualization. *Information Visualization, IEEE Symposium on*, 0:20, 2003.

[186] SYNCOPATIONSOFTWARE. DPL-faulttrees. `http://www.syncopationsoftware.com/faulttree.html`, Online; accessed 15-Jun-2012.

[187] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):61–79, 1988.

[188] A. Telea. Combining Extended Table Lens and Treemap Techniques for Visualizing Tabular Data. *Eurographics/ IEEE-VGTC Symposium on Visualization*, 2006.

[189] T. Tenev and R. Rao. Managing multiple focal levels in Table Lens. *IEEE Symposium on Information Visualization*, 1997.

[190] S. T. Teoh and K-L. Ma. PaintingClass: Interactive Construction, Visualization and Exploration of Decision Trees. In *KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003)*, 2003.

[191] I. G. Tollis, G. Di Battista, P. Eades, and R. Tamassia. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, July 1998.

[192] W. T. Tutte. How to Draw a Graph. *Proceedings of the London Mathematical Society*, s3-13(1):743–767, 1963.

[193] University of Kaiserslautern. ESSaREL. `http://www.essarel.de`, Online; accessed 15-Jun-2012.

[194] S. van den Elzen and J. J. van Wijk. BaobabView: Interactive construction and analysis of decision trees. *IEEE VAST*, pages 151–160, 2011.

[195] M. van der Borst and H. Schoonakker. An overview of PSA importance measures. *Reliability Engineering and System Safety*, 72(3):241 – 245, 2001.

[196] F. van Ham. Using multilevel call matrices in large software projects. *IEEE Symposium on Information Visualization*, 2003.

[197] F. van Ham, H. Schulz, and J. M. Dimicco. Honeycomb: Visual analysis of large scale social networks. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, INTERACT '09, pages 429–442, Berlin, Heidelberg, 2009. Springer-Verlag.

[198] J. J. van Wijk and H. van de Wetering. Cushion Treemaps: Visualization of Hierarchical Information. *IEEE Symposium on Information Visualization*, 1999.

[199] J. Vatn. Finding minimal cut sets in a fault tree. *Reliability Engineering & System Safety*, 36(1):59–62, 1992.

[200] F. Vernier and L. Nigay. Modifiable treemaps containing variable-shaped units. *In Proceedings of Extended Abstracts of IEEE Information Visualization (InfoVis '00)*, 2000.

[201] ViERforES. Virtuelle und Erweiterte Realitaet fuer hoechste Sicherheit und Zuverlaessigkeit von Eingebetteten Systemen (ViERforES). `http://www.vierfores.de`, Online; accessed 15-Jun-2012.

[202] T. von Landesberger, M. Goener, and T. Schreck. Visual Analysis of Graphs with Multiple Connected Components. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2009)*, 2009.

[203] J. Q. Walker II. A node-positioning algorithm for general trees. *Softw. Pract. Exper.*, 20(7):685–705, 7 1990.

[204] C. Ware and G. Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, 15:121–140, 1996.

[205] M. Ware, E. Frank, G. Holmes, M. Hall, and I-H. Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, September 2001.

[206] M. Wattenberg. Visualizing the stock market. In *CHI '99 extended abstracts on Human factors in computing systems*, CHI EA '99, pages 188–189, New York, NY, USA, 1999. ACM.

[207] K. Wongsuphasawat, G. Gómez, J. Alexis, C. Plaisant, T. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: visualizing an overview of event sequences (video preview). In *PART 2 - Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, pages 507–510, New York, NY, USA, 2011. ACM.

[208] J. Yang, M. O. Ward, and E. A. Rundensteiner. InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. In *Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis'02)*, 2002.

[209] Y. Yang, D. Zeckzer, P. Liggesmeyer, and H. Hagen. ViSSaAn: Visual Support for Safety Analysis. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 378–395. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

[210] Zeczer, D. Lecture: Topics in Information Visualization and Visual Analytics, University of Kaiserslautern, 2011. `http://www-hagen.informatik.uni-kl.de/~alzokari/iivva/tivva-2011.html`, Online; accessed 15-Jun-2012.

[211] J. Zhang, L. Gruenwald, and M. Gertz. VDM-RS: A visual data mining system for exploring and classifying remotely sensed images. *Computers & Geosciences*, 35(9):1827–1836, sep 2009.

[212] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05. IEEE Computer Society, 2005.

[213] C. Ziemkiewicz and R. Kosara. Embedding Information Visualization within Visual Representation. In Z. W. Ras and W. Ribarsky, editors, *Advances in Information and Intelligent Systems*, volume 251 of *Studies in Computational Intelligence*, pages 307–326. Springer Verlag, 2010.

[214] G. Zipf. Computation of minimal cut sets of fault trees: Experiences with three different methods. *Reliability Engineering*, 7(3):159–167, 1984.

Fault Tree Symbols

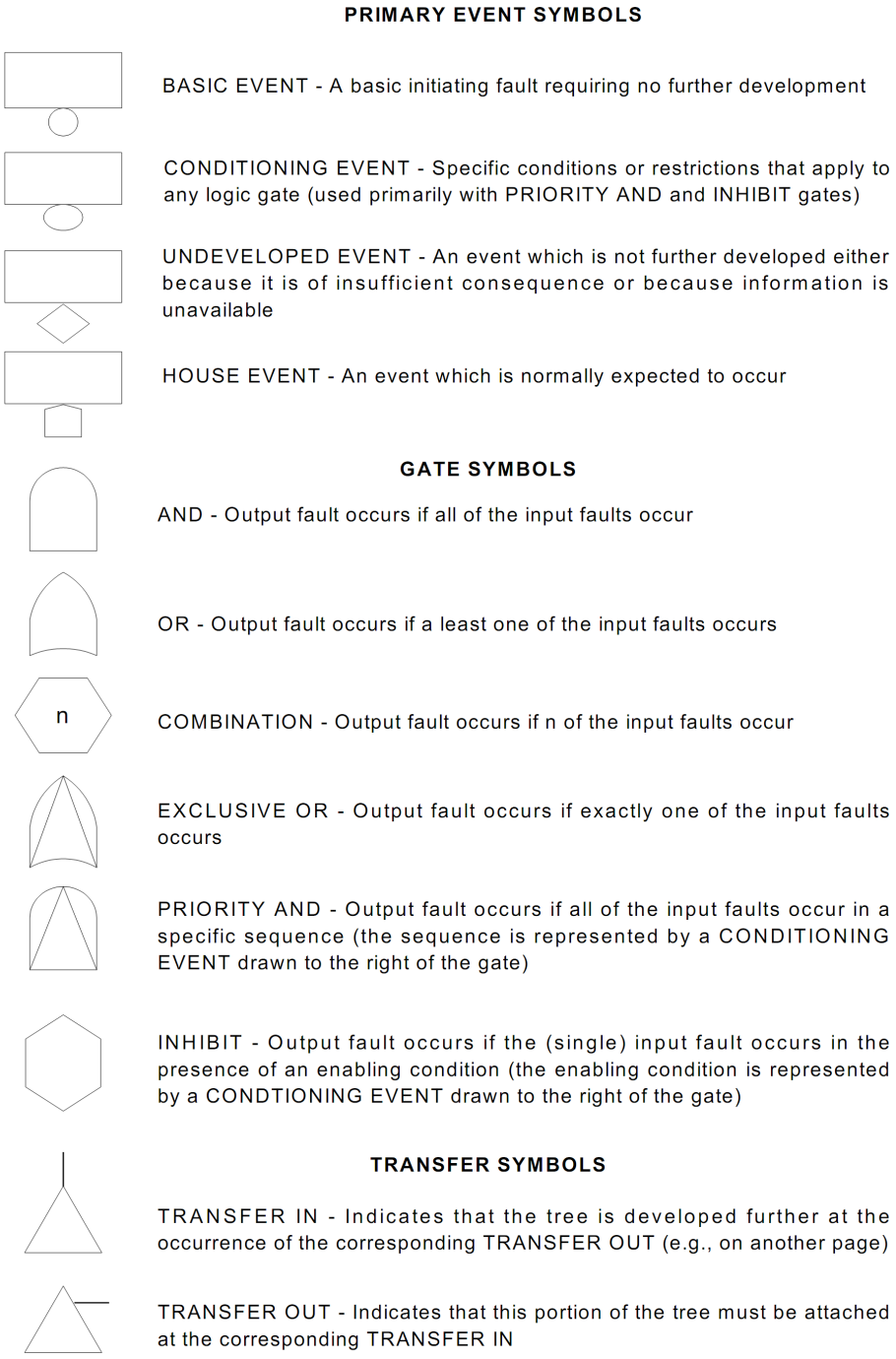


Figure 9.1: Fault tree symbols (US style) (produced by [142]).

List of Publications

- Yi Yang, Patric Keller, and Peter Liggesmeyer. Visual Approach Facilitating the Importance Analysis of Component Fault Trees. *3rd International ACM/GI Workshop on Digital Engineering (IWED) in conjunction with SAFECOMP 2012*, Magdeburg, Germany, 2012.

- Yi Yang, Patric Keller, Yarden Livnat, Peter Liggesmeyer. Improving Safety-Critical Systems by Visual Analysis. In Christoph Garth, Ariane Middel, and Hans Hagen, editors, *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011*, volume 27 of *OpenAccess Series in Informatics (OASIcs)*, pages 43-58, Dagstuhl, Germany, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

- Yasmin I. Al-Zokari, Yi Yang, Dirk Zeckzer, Peter Dannenmann, and Hans Hagen. Towards Advanced Visualization and Interaction Techniques for Fault Tree Analyses Comparing existing methods and tools. *PSAM11 (Probabilistic Safety Assessment) & ESREL (European Safety and Reliability Association) 2012 Conference*, Finland on 25-29 June, 2012.

- Peter Liggesmeyer, Henning Barthel, Achim Ebert, Jens Heidrich, Patric Keller, Yi Yang and Axel Wickenkamp (2012). Book chapter: Quality Improvement Through Visualization of Software and Systems, *Quality Assurance and Management*, Prof. Mehmet Savsar (Ed.), ISBN: 978-953-51-0378-3, InTech.

- Yi Yang, Dirk Zeckzer, Peter Liggesmeyer, and Hans Hagen. ViSSaAn: Visual Support for Safety Analysis. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors, volume 2 of Dagstuhl Follow-Ups*, pages 378-395. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

Curriculum Vitae

Personal Information

Name:	Yi Yang
Place of Birth:	Shandong, China
Nationality:	Chinese

Education

Nov.2008 – Oct.2012 **University of Kaiserslautern** **Kaiserslautern, Germany**

- PhD study in Computer Science
- Dissertation: Visual Support for Safety Analysis

Apr. 2001 – Sep. 2007 **RWTH Aachen University** **Aachen, Germany**

- Diplom study in Computer Science
- Degree: Diplom-Informatiker
- Diplom thesis: Analysis of source code with Latent Semantic Indexing

July 2000 - Apr. 2001 **Language school *Sprache Akademie*** **Aachen, Germany**

- German course

Sep.1996 - June 1999 **Shandong University of Science and Technology** **Jinan, China**

- College, major: Information system for finance

Sep.1993 – June 1996 **Attached Senior School of Shandong Normal University** **Jinan, China**

- Senior high school

Sep.1990 – July 1993 **Jinan NO.5 High School of Shandong Province** **Jinan, China**

- Junior high school

Working Experience

Dec.2007 – Oct.2008 **Achievo Corporation** **Beijing, China**

- Software requirement analyst

Dec.2006 – May 2007 **RWTH Aachen University** **Aachen, Germany**

- Software developer (working student)

2004 – 2005 **Quanta Computer Company** **Aachen, Germany**

- Repairer of laptop computers (Student job)

2004 **RWTH Aachen University** **Aachen, Germany**

- System administrator (working student)

June 1999 - July 2000 **China Shandong International Economic & Technical Cooperation Corp.** **China**

- Accountant